ESD-TR-75-352

MTR-3071

MTI PROCESSOR SOFTWARE

FEBRUARY 1976

Prepared for

DEPUTY FOR SURVEILLANCE AND NAVIGATION SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Bedford, Massachusetts

Approved for public release;
distribution unlimited.

Project No. 6290
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-76-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

John E. Dunkle

JOHN E. DUNKLE, Major
Project Manager

James E. Troutman

JAMES E. TROUTMAN, 1Lt
Project Engineer

FOR THE COMMANDER

Roy F. Pollino

ROY F. POLLINO, GS-15
System Program Director
Weather Instrumentation SPO
Deputy for Surveillance and Navigation Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-75-352	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* MTI PROCESSOR SOFTWARE		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER MTR-3071
7. AUTHOR*(s)* D. D. Coffin H. E. T. Connell		8. CONTRACT OR GRANT NUMBER*(s)* F19628-76-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation Box 208 Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 6290
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Surveillance and Navigation Systems Electronic Systems Division, AFSC Hanscom Air Force Base, Bedford, MA 01731		12. REPORT DATE FEBRUARY 1976 13. NUMBER OF PAGES 250
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

DIGITAL MTI
DISTRIBUTED PROCESSING
ELECTRO-OPTICS

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The digital "snapshot" moving target indication (MTI) technique was developed for the GEODSS project in FY-75. GEODSS is a Ground-based Electro-Optical Deep Space Surveillance system for the detection of satellites.

This document describes the software developed to determine the feasibility of performing snapshot MTI on a multi-minicomputer processor (MMCP). The

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Date Entered)*

19. Key Words (Continued)

GEODSS Ground-based Electro-Optical Deep Space Surveillance System
MINICOMPUTERS
MULTI-COMPUTER NETWORKS
MULTI-COMPUTER SOFTWARE
MULTI-MINICOMPUTERS
NETWORK SOFTWARE
NETWORKS
REAL-TIME PROCESSORS
SNAPSHOT MTI

20. Abstract (Continued)

software resides on eight interconnected Data General Corporation 800 series minicomputers and constitutes a reasonable base from which an operational system may be developed.

ACKNOWLEDGMENTS

This report has been prepared by The MITRE Corporation under Project No. 6290. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

The software described in this document represents an implementation of a scheme conceived by J. E. Barry of The MITRE Corporation. His guidance and inspiration are gratefully acknowledged.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

GLOSSARY

CAM	Cancellation Minicomputer
CLI	Command Line Interpreter, a feature of RDOS
CRT	Cathode Ray Tube
DGC	Data General Corporation
DIM	Disc Interface Minicomputer
FOV	Field of View
GEODSS	Ground-based Electro-Optical Deep Space Surveillance System
MCA	Multiprocessor Communications Adapter
MCABOOT	An RDOS CLI Instruction
MTI	Moving Target Indicator
PIM	Preprocessor Interface Minicomputer
RDOS	Real Time Disk Operating System
RECM	Reconstruction Minicomputer
TRKIN	Track Initiation Minicomputer

1.0 INTRODUCTION

The digital real-time "snapshot" MTI was developed for the GEODSS project in FY 75. GEODSS is a Ground-based Electro-Optical Deep Space Surveillance System. This document describes the software; an accompanying document, ESD-TR-75-351, is a description of the multi-minicomputer system on which this software was implemented. The software resides in eight inter-connected minicomputers. The algorithms have been implemented in a configuration suitable to evaluate the feasibility of performing snapshot MTI in real time by using the computers cooperatively. Ultimate conversion of these programs to function in a real-time system has been considered in the design process.

The snapshot MTI technique described here is a digital computation method which can be used to discover a satellite that moves slowly through a field of stars. The GEODSS observation system employs a telescope to obtain precise visual images of portions of the night sky. The images are focused onto a light-sensitive storage tube to enable a conversion of visual images to electronic images. The storage tube is systematically scanned by an electron beam to effect a read out of the tube. The electron beam current is compared to a threshold to determine the presence of image points on the storage tube. The X, Y coordinates of the electron beam on

the storage tube along with an indication of amplitude comprise the data representing each detection. A preprocessor conditions these detections to remove the multiple detections one obtains as the beam scans across a single "point" image. The entire group is simply replaced by a single representative point.

The resulting picture is a "snapshot" of the telescope field of view and is the input to the MTI processor. A sequence of snapshots is similar to the individual frames of a motion picture. A time sequence of frames from the same field of view may be compared to detect the motion of a point image.

The snapshot MTI algorithm as implemented here performs a frame-to-frame cancellation of corresponding stars. The remainder or "leakers" are treated as potential moving target positions. Any leakers which seem to progress linearly with time over 3 frames may be labelled a target, presumably for further attention by a tracking device. The tracker may be vectored to some later point along the course indicated by the linear track.

2.0 THE SNAPSHOT MTI IMPLEMENTATION

2.1 General

This section describes the structure for the snapshot MTI demonstration software. Figure 1 shows the functional elements of the multiprocessor software. Each element resides in a separate minicomputer. Inter computer data transfer is accomplished over the MCA data bus which is common to all processing elements.

The digital representation of the telescope field of view (FOV) enters the MTI processor from the preprocessor to the PIM (Preprocessor Interface Minicomputer). In the demonstration program, the entire frame of data is passed on directly to the DIM (Disc Interface Minicomputer). The PIM also determines the bounds within the frame that are necessary to distribute portions of the FOV to each cancellation computer. The raster scan of the FOV is split into 4 strips of equal area by examination of the y coordinate (Figure 2). All detections in each strip are sent to the proper CAM to serve as a reference frame. Each CAM functions as a Cancellation Minicomputer.

The DIM also splits the FOV into 4 equal area strips for distribution to the CAMS after applying a digital threshold. The digital threshold is read from the DIM computer keys on its own operator's console. This function is available to facilitate examination of the sensitivity of the multiple threshold cancellation. It is expected that an operational

IA-46,537

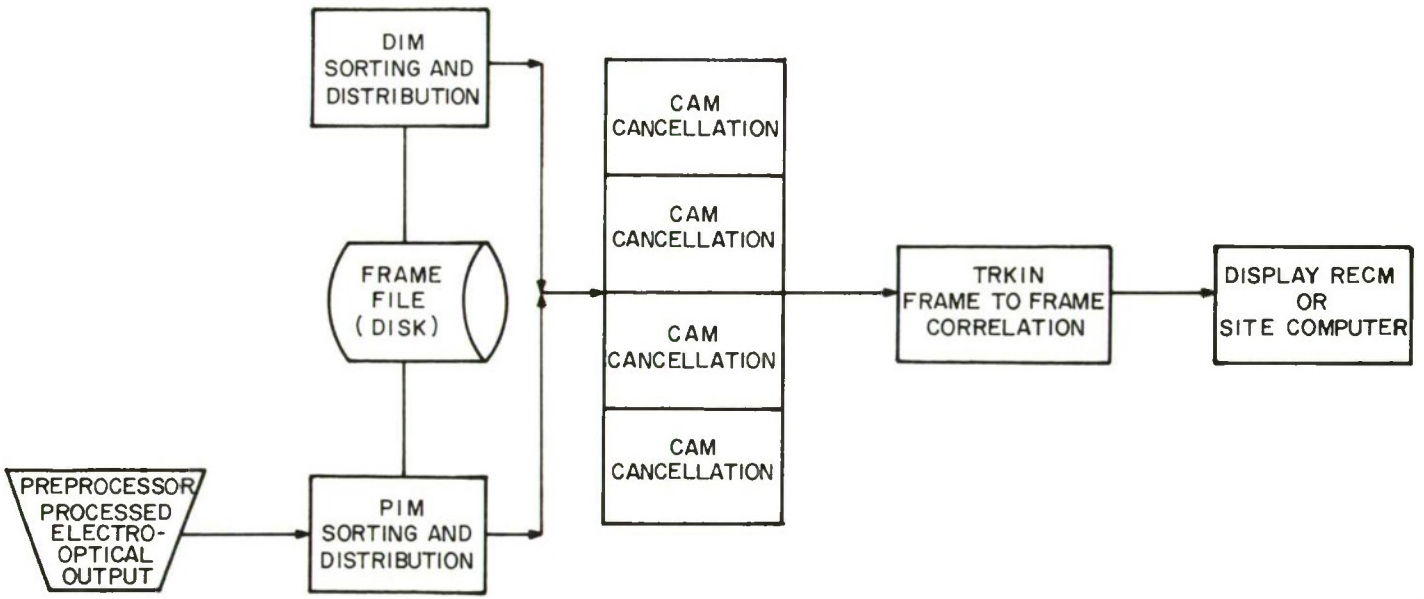


Figure 1 SNAPSHOT MTI SYSTEM ELEMENTS

IA-46,538

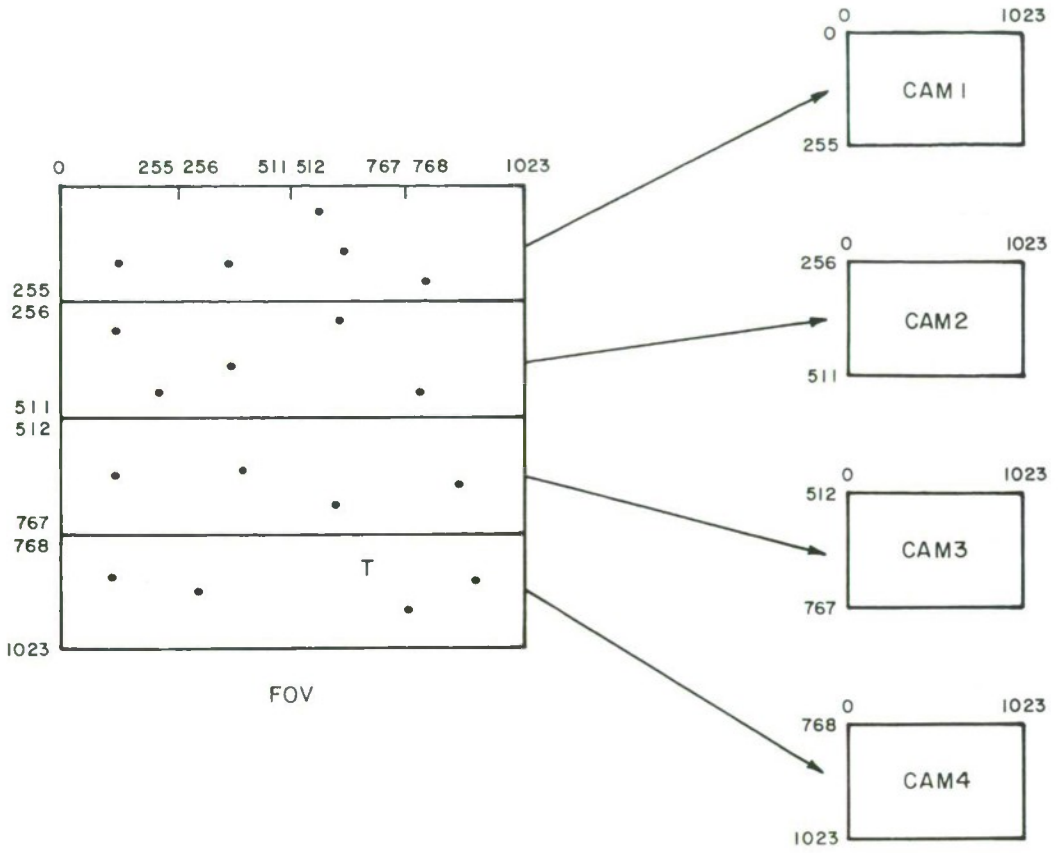


Figure 2 FIELD OF VIEW STRIP SUBDIVISIONS

version of the program would either have a preset value in the program or would have a base value that would be varied if the number of MTI system false alarms became excessive. All data points above the digital threshold in each strip are sent to the proper CAM.

Each CAM compares the data strip it receives from the DIM with the next strip it receives from the PIM. In this way a delay of one PIM frame is achieved, so that target motion from frame to frame will result in a leaker or uncancelled data point. The CAM cancellation applies a one cell fringe around each reference frame data point to accomodate detections near the edge of a resolution cell. The cancellation process forms a list of all those data points that are in the DIM frame (at the higher threshold) but not in the PIM frame. The finished list of "leakers" is sent by each CAM to a reconstruction and Track Initiation Minicomputer (TRKIN) for storage.

The TRKIN receives and re-assembles leakers from all CAMS and saves them as a "leaker frame". Three such leaker frames are accumulated to perform the track initiation function. The data points in the first and third frames are paired in all combinations. Each pair is used to compute the expected coordinates of a data point in the second frame assuming linear motion and frames equally spaced. If there is a data point in the second frame that corresponds to the computed point, a track message is generated. The track message consists of matching the first and third frame coordinate pairs and the entire second frame of leakers.

The PIM, CAMs and TRKIN all send their output data to the display RECM (display Reconstruction Minicomputer) to enable a visual presentation of various stages in the MTI process. This display is an x, y CRT. Any one of the PIM, CAM or TRKIN outputs may be held on the display.

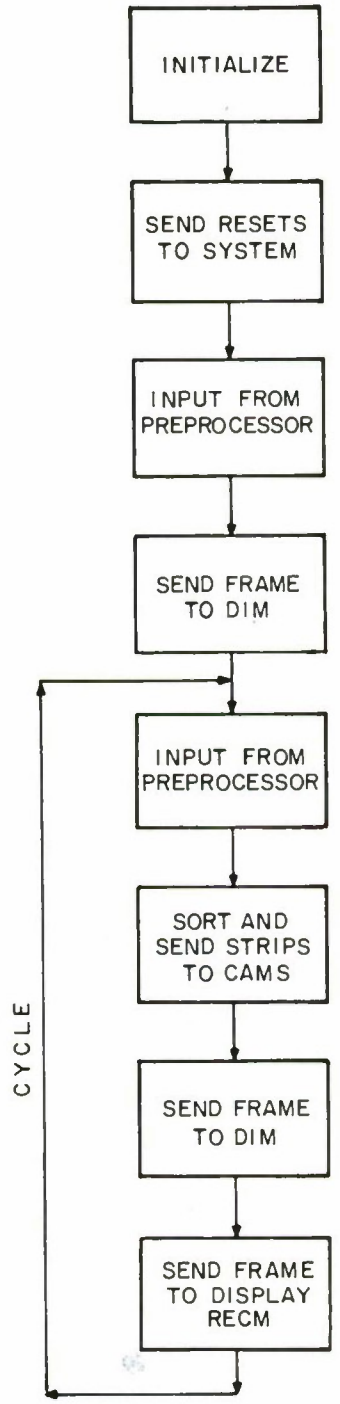
2.2 Inter Computer Coordination and Communication

This section describes the principles and techniques used to achieve cooperation between the processing elements. Messages, coordination, and data transfers all occur over a single data bus capable of a 300 kHz word transfer rate. This description deals with the overall structure, the software interface, and the hardware interface, upon which the software is heavily dependent.

2.2.1 MTI Software Control

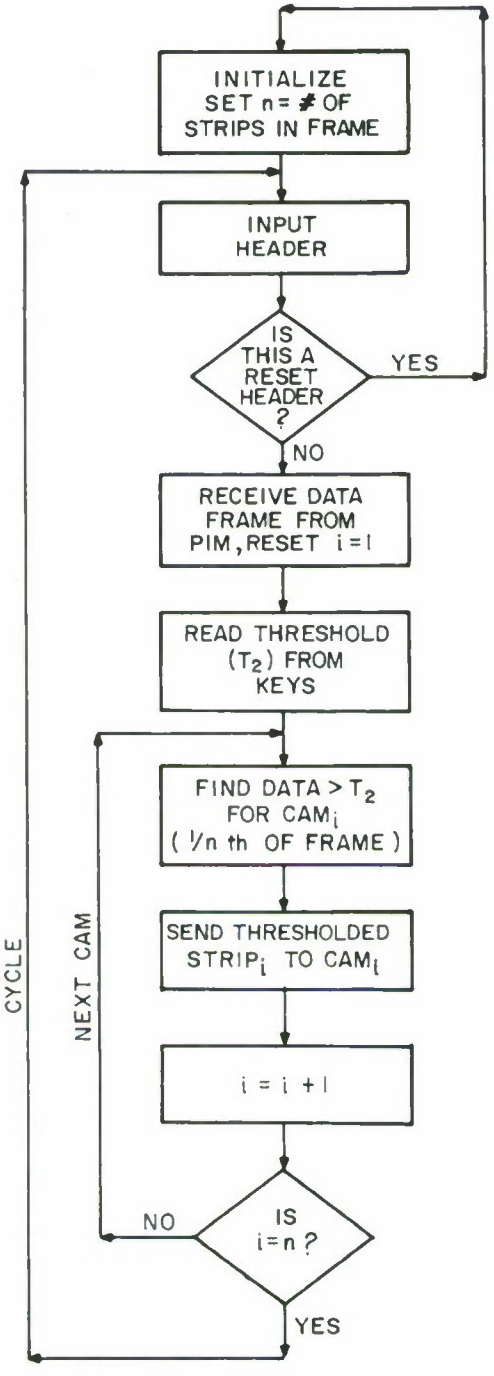
Each computer main program has its own MTI sub-cycle which must be completed (or aborted) during the overall MTI cycle. Figures 3 through 6 show simplified representatives of the cycle logic for each program. The cycle must be accomplished in the time period defined by the read out cycle of the star field from the preprocessor. Depending on the character of the data, the quantity of data and distribution of data in the field of view, the actual time required by segments of the MTI processing chain or branches will vary considerably.

The approach that has been taken is intended to take advantage of speed whenever it occurs, to accommodate sluggish segments, and



IA-46,539

Figure 3 PIM PROGRAM FUNCTIONAL FLOW



IA-46,540

Figure 4 DIM PROGRAM FUNCTIONAL FLOW

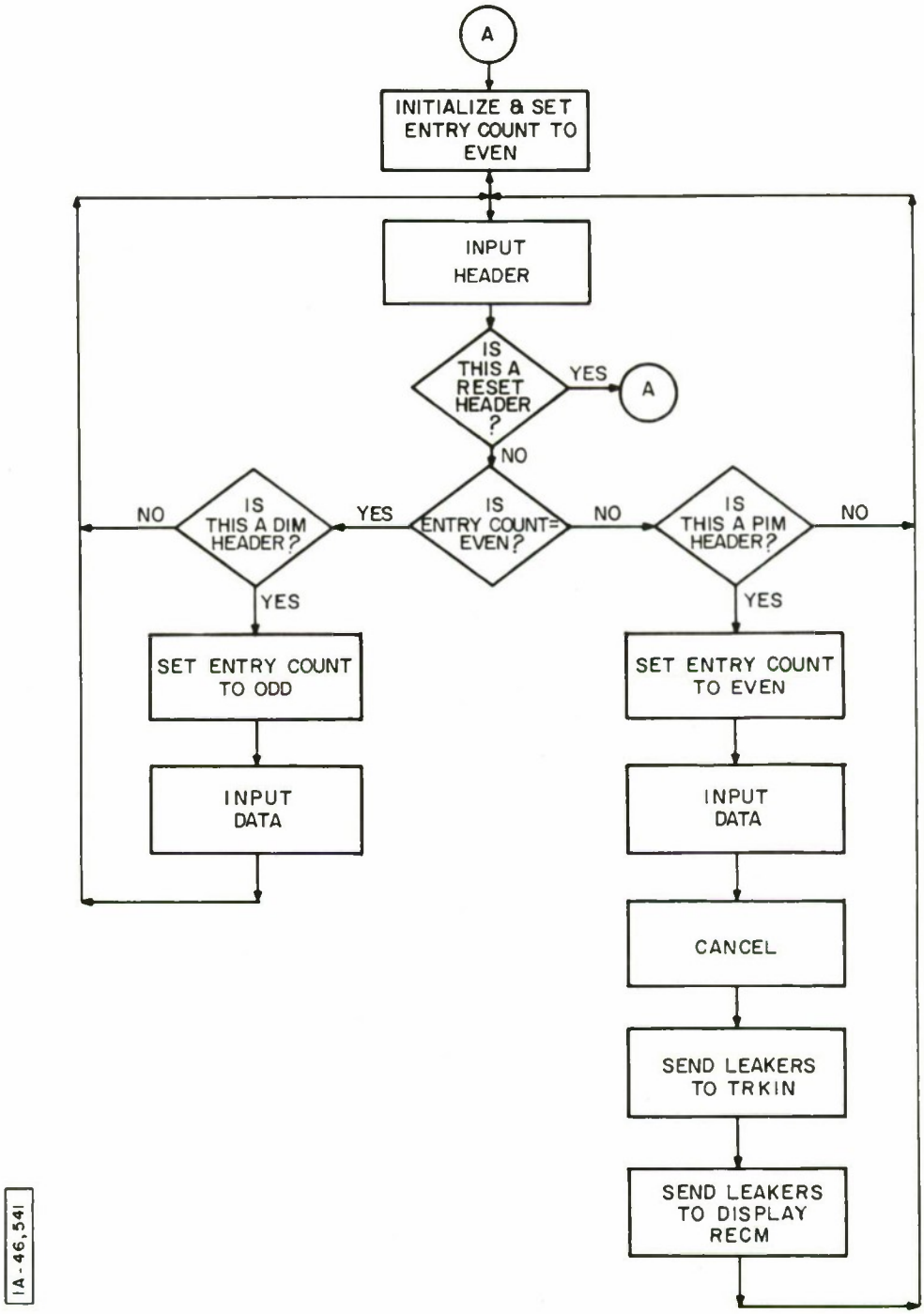
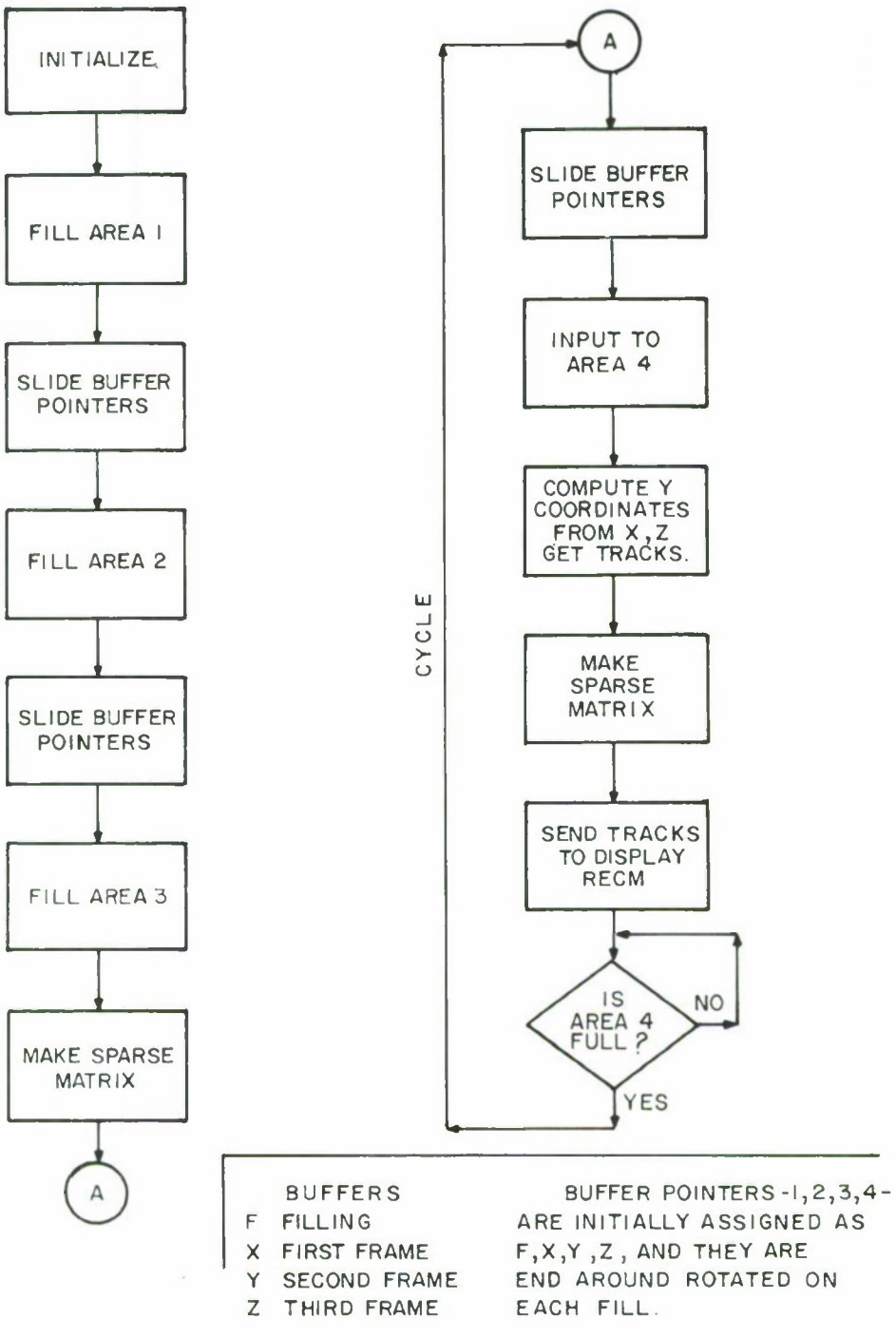


Figure 5 CAM PROGRAM FUNCTIONAL FLOW



IA-46,542

Figure 6 TRKIN PROGRAM FUNCTIONAL FLOW

to adjust for radical variations. A typical sequence of events is depicted in Figure 7.

The logic of the main program in each computer controls the sequence of events permissible to that program. This includes the inputting of data from any other computers. Clearly each computer cannot know a priori when to expect data from other computers since the computer clocks are asynchronous and the data loads vary. In order to insure the proper overall sequence of events in the multi-computer system, then, the software procedure is to accept messages from other software. These messages or Header Blocks describe the transfer of data that is desired by the outputting machine. Each Header message that does enter a machine must be answered in order to achieve a software handshake. The answer or Acknowledge Block can indicate either software acceptance or refusal of the intent implied by the Header. A refused Header is simply resubmitted as soon as possible by the rejected computer. This assumption implies that a Header is only refused because it is premature or out of phase. To guarantee that this can in fact be the only reason for refusal, the data paths through the multi-computer system are rigid. That is, even if no data transfer is needed, say, as the result of a perfect cancellation, the cancellation computer must send a Header describing that zero block and the zero block itself. To guarantee that no block comes too late, it is required that all blocks are accounted for in each program MTI cycle before the program may proceed to the next cycle.

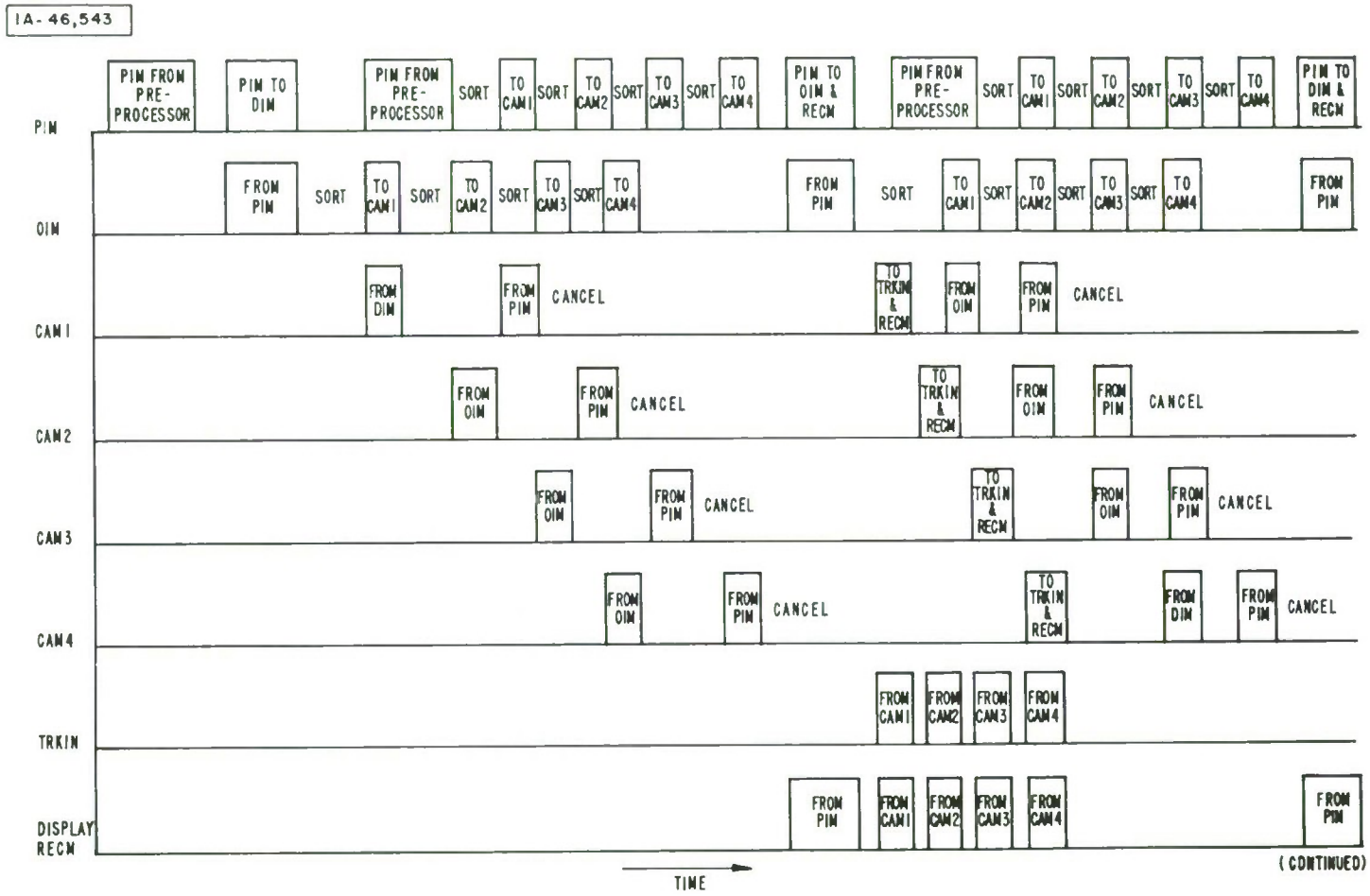


Figure 7 MTI TYPICAL EVENT SEQUENCE

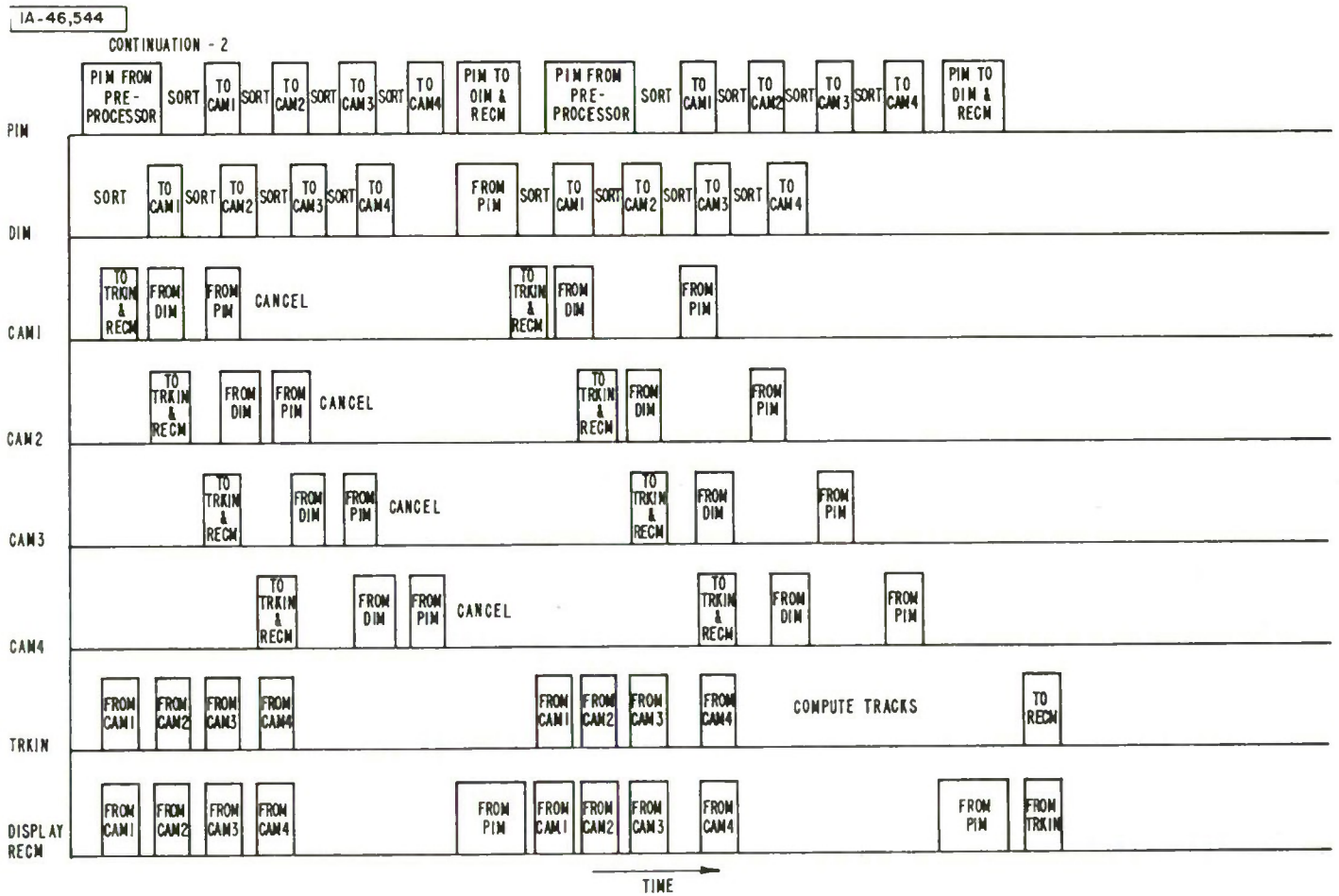


Figure 7 MTI TYPICAL EVENT SEQUENCE

The rigid sequence requirements of the computer coordination scheme are only between MTI program cycles. Consider, for example, the reconstruction of the FOV leakers from the cancellation computer. At the start of each cycle, the program makes a list of the cancellation computers. As the CAMs respond, they are deleted from the list, thereby enabling any response sequence to occur. If one is heard from but is not in the list, it is assumed that that Header is indeed from a legitimate computer, but that it is ahead of schedule. The negative Acknowledge of the Header is interpreted as described above and that CAM keeps trying until it is successful.

The normal state of all programs which have either finished their MTI cycle or have finished with their input data area is to listen for headers. This condition can be used to effect an MTI system reset. A Reset Header has been defined to serve this purpose. The restarting of the PIM causes the Reset Header to be sent to all computers and thus initialize the MTI sequence.

2.2.2 The MCA Bus

The functional behavior of the Data General Multiprocessor Communications Adapter (MCA) Data Bus is an integral part of the logic used to coordinate the inter computer transfers. In effect it allows for virtual ignorance of the condition of the computer intended as the destination for an information transfer. In fact, several machines may be stacked up waiting for access to one computer

while that computer is busy without causing any difficulties. This description, then, is necessary for one to comprehend the program logic of this multi-computer software.

The MCA bus consists of data, address, control and timing paths. These are connected to the MCA transmitter and receiver controllers in each computer. The controller in each computer is given a unique four bit address by a positioning of jumper wires. Timing for the entire bus is supplied by the controller of one computer. This computer is chosen by necessity to be at one end of the bus; the computer at the other end of the bus is used to indicate that a bus clock cycle may be completed and all computers have been attended.

The controllers each contain typical block transfer logic; that is, a register containing a pointer to a specific core location, a word transfer count register, a data register and a status register.

To begin a data transfer, the MCA controller pointer and count registers are set by the computer program and the controller is activated. The transmitter controller steals a computer cycle to acquire data for the data register. The next bus clock cycle (originating from the "right end" controller), which arrives at the transmitter and for which the bus status is not busy, causes the data and the transmitter and destination addresses to be placed on the bus. This action also sets the bus busy condition for the duration of this clock cycle, preventing other transmitters from using the clock cycle.

The clock, control signals, data, transmitter and destination travel down the bus until the end is reached. On reaching the end of the bus, the "return" control signal is set, and the receivers respond as it reaches them. If these signals are intercepted by a receiver controller that is unlocked, active and has an address that matches the destination, the data word is accepted and an echo is "sent back" to the transmitter before the clock cycle closes the bus. This echo is used by the transmitter to update its word count and address pointer.

If the receiver is not active, locked to another transmitter or powered down, the "echo" line is not set.

The overall behavior of this bus is such that a program can initiate a transmitter block transfer and then proceed, even if the intended receiver is busy. The transmission will occur whenever the receiver becomes "unbusy" and active. Further, a receiver can be turned on at will by the program, regardless of whether a transfer has been already requested by a transmitter. The effective use of this bus by the MTI software for inter computer transfers and computer program phasing is a matter of intelligent software acceptance or rejection of data.

2.2.3 Data Communication Structure

A computer ready to transmit data to another computer proceeds as shown in Figure 8. The 16 word Header Block described in Appendix III

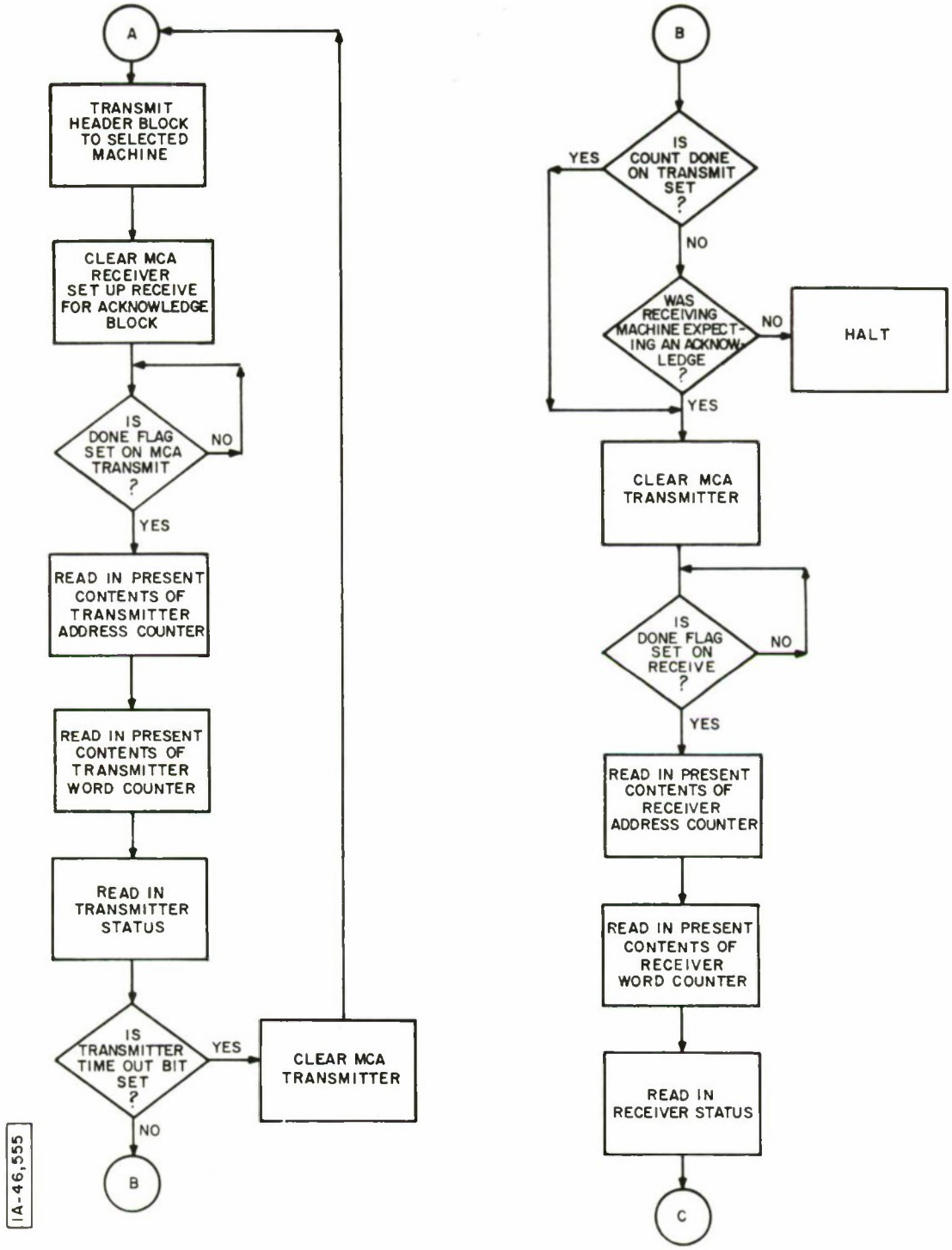


Figure 8 GENERAL HEADER TRANSMISSION, ACKNOWLEDGE RECEPTION AND DATA TRANSMISSION

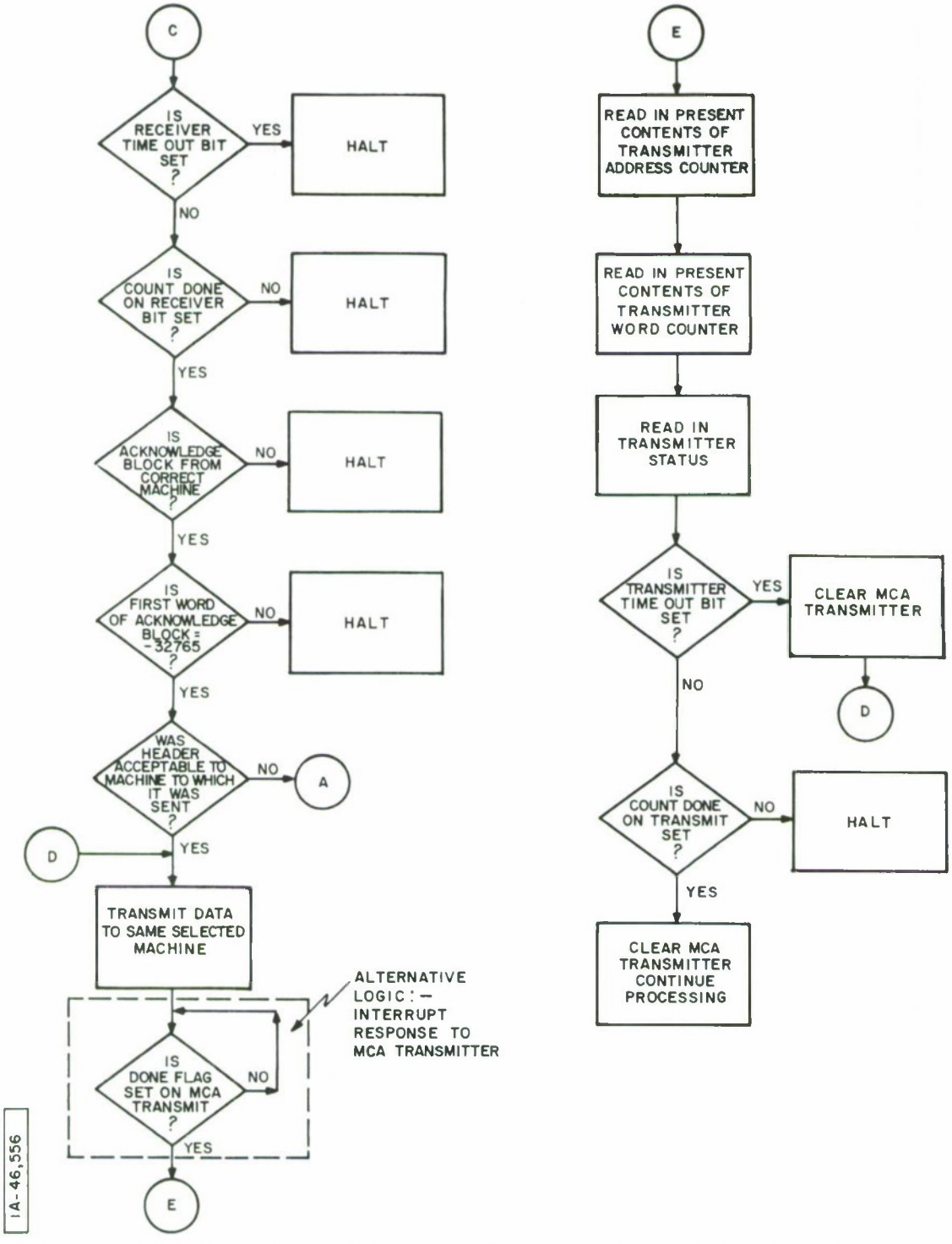


Figure 8 GENERAL HEADER TRANSMISSION, ACKNOWLEDGE RECEPTION AND DATA TRANSMISSION - 2

is prepared and a block transfer to the destination is initiated on the bus. At this point the program demands a successful hardware completion of that transfer and a coherent response by the software in the other machine. The program must not proceed beyond this stage if the transfer fails. The demonstration version program simply halts if conditions indicate a hopeless outcome. The operational programs would instead revert to waiting for a header if all else fails in the hope that a reset header will come along to restart the process.

As soon as the two computers establish a Header-Acknowledge contact that is acceptable, the access to this communication link is kept locked from all other computers. The data transfer is accomplished during this time; and upon its successful completion, the transmitter signs off.

A computer prepared to receive data from some other computer proceeds as shown in Figure 9. Any inputs are checked for block type. Headers are the only acceptable transfers at this stage. Given that a header has found its way into the receiver, the program must check a list of machines from which some response is expected. As each machine responds, it may be deleted from the list. One machine may be allowed to have multiple entries in the list to enable such occurrences without complicating the check-off logic. This list should be reset on each MTI cycle.

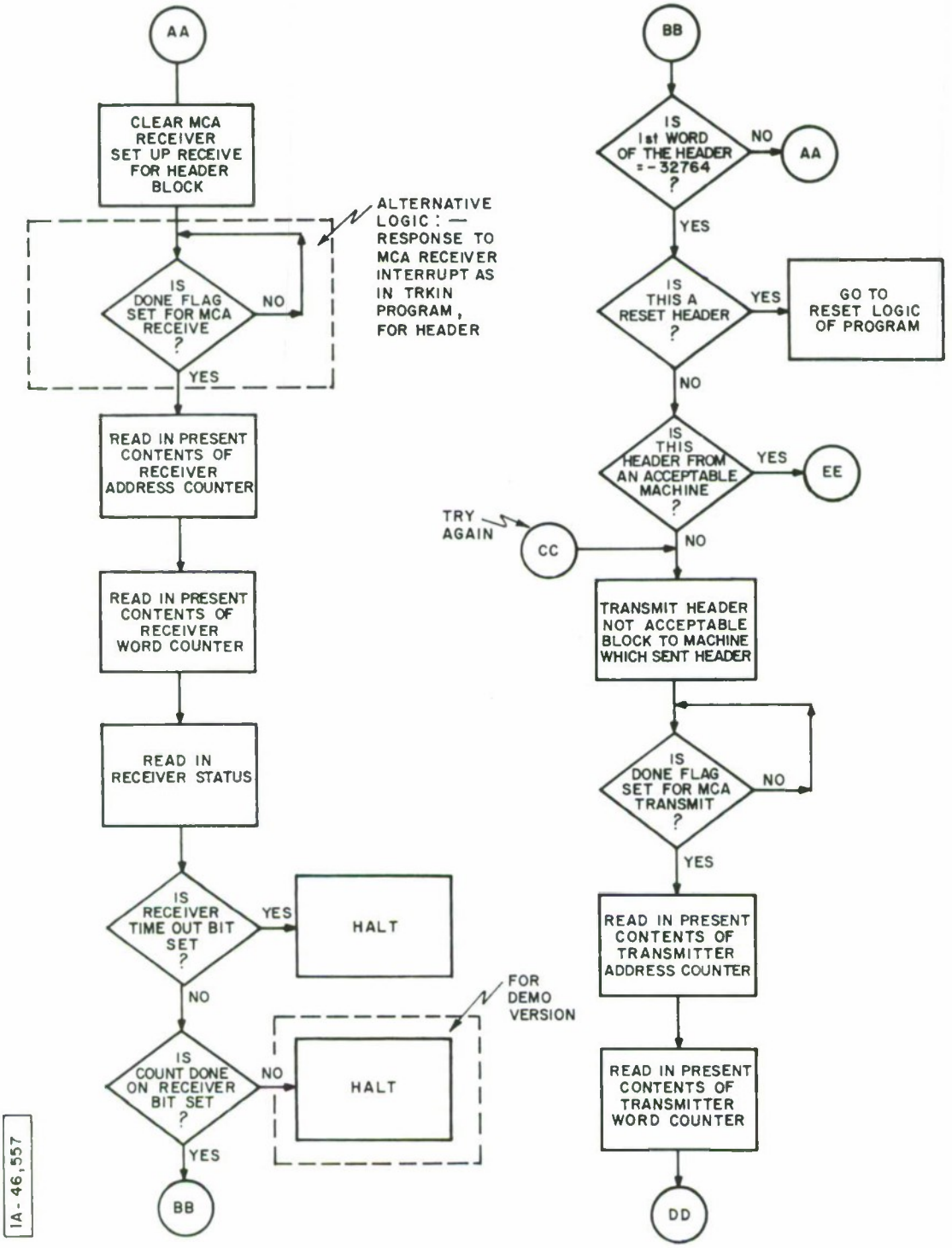


Figure 9 GENERAL HEADER RECEPTION, ACKNOWLEDGE TRANSMISSION, AND DATA RECEPTION

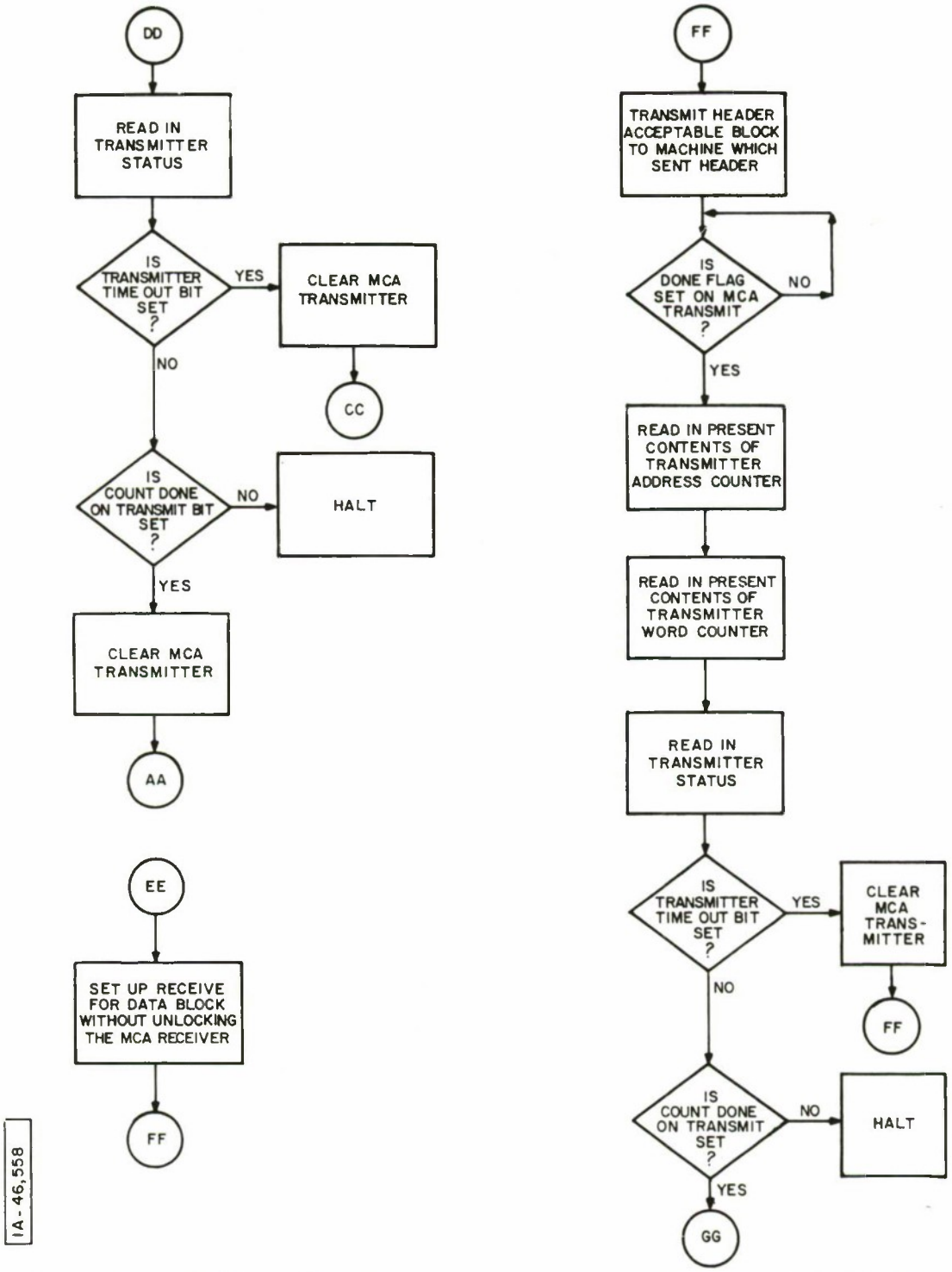


Figure 9 GENERAL HEADER RECEPTION, ACKNOWLEDGE TRANSMISSION, AND DATA RECEPTION-2

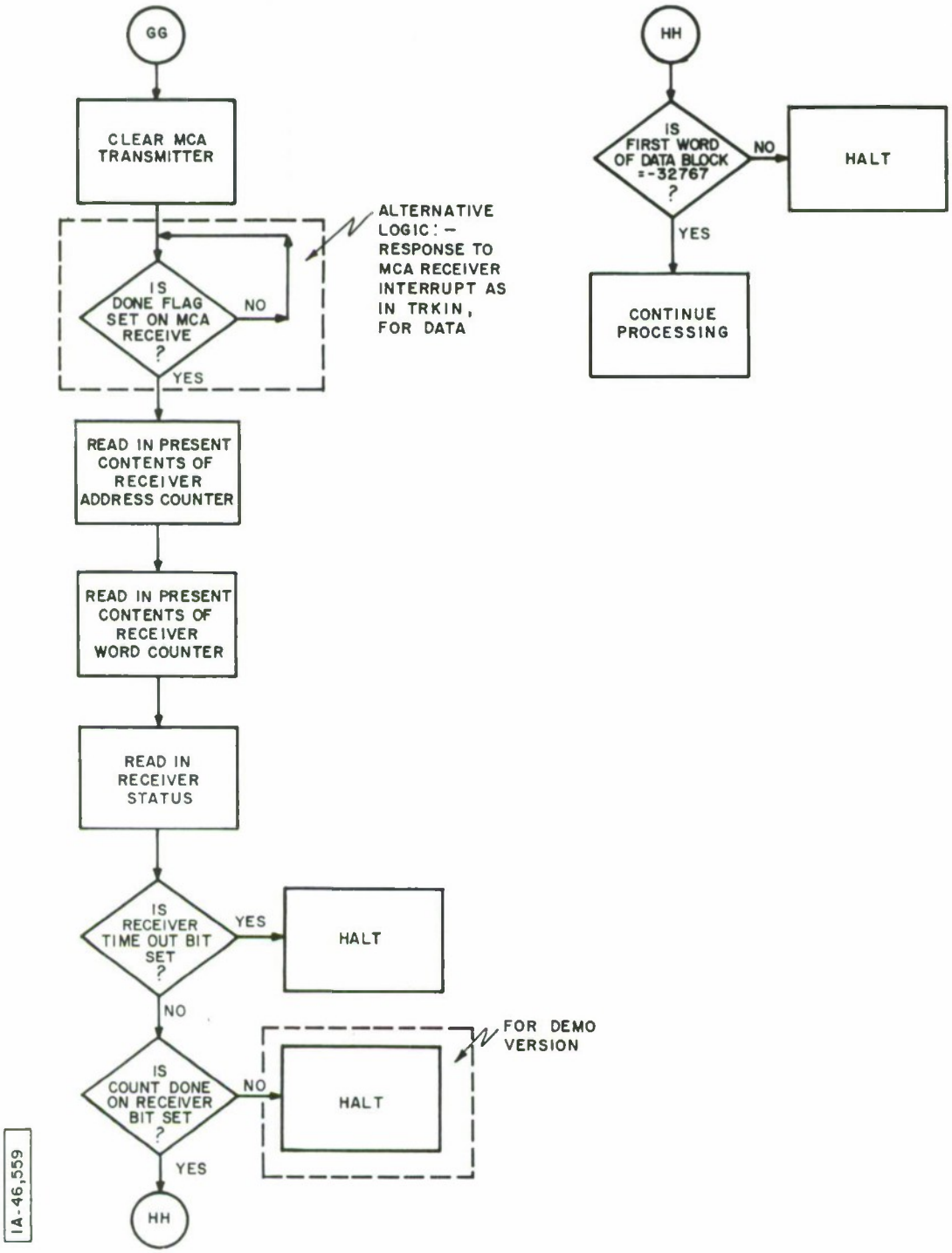


Figure 9 GENERAL HEADER RECEPTION, ACKNOWLEDGE TRANSMISSION, AND DATA RECEPTION - 3

The acceptability or non-acceptability of a header is indicated by the acknowledge block. Presumably a negative acknowledge means that the program may resume listening for any header. A positive acknowledge means data will follow as described in the header. Header descriptions, however, are not allowed to exceed buffer sizes.

2.3 Cancellation Method

The digital representation of the FOV for communication between computers is in the form of a list of x, y coordinates. This is the most compact data form for such transfers and minimizes the amount of time used. For cancellation, however, sorting of lists to find corresponding entries is very time consuming due to the multiple access required of each list entry. In order to compare two pictures as is done to achieve cancellation, it is also time consuming if one must examine each picture resolution element in both pictures. In fact, for the picture represented by 1024 by 1024 pixels, there are 130,000 words to examine, if the picture elements are allotted 1 bit each, that is, if one 16 bit computer word carries 16 pixels of information. This representation is referred to as the sparse matrix picture.

The sparse matrix picture does have considerable advantage when two pictures are to be compared, provided at least one picture is available in list form. With one picture in list form, one can retrieve the specific 16 bit group of pixels containing the pixel of interest by a simple calculation relating pixel coordinates from the list to matrix coordinates in the array.

This technique will be called sparse matrix list processing. It effects a considerable savings in processing time.

The cancellation process then is to accept two lists for comparison. The first list is converted to a sparse matrix. The second is the check list to be cancelled.

In order to avoid the requirement to store the complete sparse matrix presentation of the strip to be cancelled in any computer, we take advantage of the fact that the list data is y-ordered. That is, as one scans down a list of coordinates, one encounters an ordinal sequence of x values until y changes. The y changes are also ordinal, but vary less rapidly. This fact is used by noting the value of y (but only when it changes) as compared to some arbitrary bound. The sparse matrix up to that bound may be generated and retained while processing is then swapped to the other list involved in the comparison. This list is processed until the same bound is reached. The list swapping proceeds until the y bound corresponds to the end of the data for the strip. The overall effect of swapping back and forth between the two lists appears effectively as a window or sub strip that steps down through the pixel matrix (figure 10).

The bounds used to determine the matrix size which may exist in a computer at one time and the y extent of each list are tailored to match the binary form of the x, y coordinates. Computation of matrix position from a list, then, degenerates to a masking and

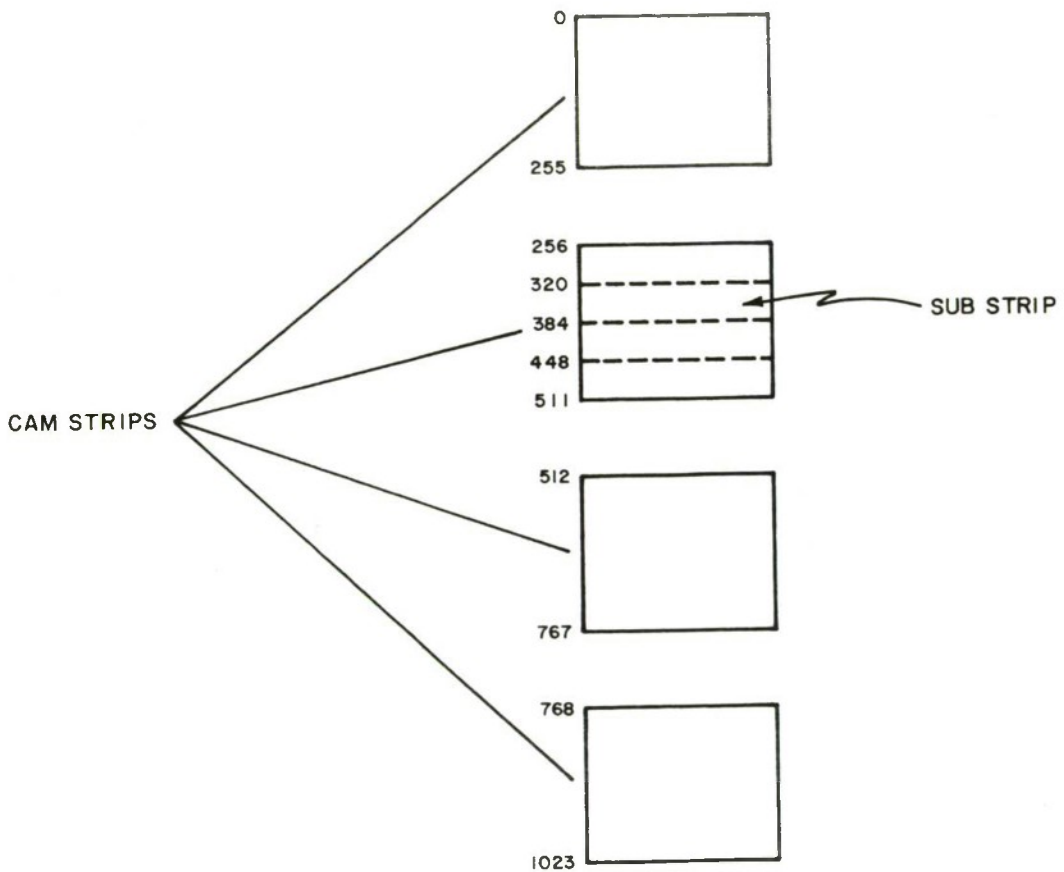


Figure 10 SUB STRIP SPARSE MATRIX

indexing operation which can be done very efficiently by the computer.

To summarize, first the sparse matrix is cleared. The reference frame list is then converted to the bounded sparse matrix with a bit pattern that represents the sphere of influence of each "illuminated" pixel. In this version of the program the bit pattern includes the illuminated bit and all abutting pixels to prevent borderline leakers. These pixels are inclusive ored with any bits already set in the sparse matrix. On reaching the bound, the frame to be cancelled is used to generate the coordinates of the illuminated pixel in the same sparse matrix. Anding this single bit pattern with the sparse matrix entry then enables a decision of whether or not to declare the x, y point in hand a leaker.

When the y bound in this list is reached, the next bound is selected, and processing again transfers to the reference list after first clearing the matrix. When the final bound is reached, the cancellation process is complete for the frame at hand and the leakers may be passed along to the next computer.

2.4 Track Initiation

A track is initiated by noting linear or near linear motion of an illuminated spot on three successive frames of leakers. We note that the motion is best determined by using the first and last frames to compute the position on the middle frame. The position error is $\frac{\sqrt{2}\,\sigma_r}{n\Delta t}$ where σ_r is the positioning error within one frame,

Δt the spacing between frames, and n the number of frames spanned by the estimate. Using the end frames has the very interesting additional feature that all spot pairings have an observable coordinate on the middle frame, thereby avoiding complicated tests. This approach to comparison is based on an interpolation and not an extrapolation and thereby avoids error growth.

This method for comparison is similar to that used for cancellation, that is, the sparse matrix method of list processing. The primary difference is that the entire sparse matrix is generated. However, to accommodate possible observation errors the matrix is "defocused." The defocusing is achieved by ignoring the two least significant bits in each coordinate. The central frame list of leakers are converted to the defocused sparse matrix in much the same manner as the star field was for cancellation. The combinations of 1st and 3rd frame pairs are then converted to sparse matrix coordinates for anding with the sparse matrix entry. A comparison gives the potential track list. The track list is then appropriately displayed on a monitor and the 1st, 3rd points track vector and the frame identification will be sent to the site computer for verification with a file of resident space objects (RSO's).

3.0 COMPUTER PROGRAM DISTRIBUTION

This section describes the technique used to bring the MTI software up from a cold start. It is assumed that one can use the standard Data General Corporation Real Time Disk Operating System (RDOS) and that both RDOS and the MTI software reside on the primary fixed head disc. The general approach is to turn on all computers with a so-called rim loader program running. In this case the rim loader is a two instruction machine language program that is either manually keyed into the computer or loaded in a shorter keying sequence with computers which have the hardware "Automatic Program Load" option. The rim loader causes each computer to expect a specific program code sequence over the MCA bus.

There is an RDOS procedure which also resides on the disc, that causes each MTI program to be sent to a prearranged computer. The programs automatically begin execution when they arrive in each machine. The final step is to "leave" RDOS and enter the MTI program PIM which takes its place in the main computer.

In the operational version of the PIM (and other programs), a brief question, answer and assignment sequence between the programs precedes the actual MTI program execution. This conversation determines which programs are in which machines and informs all programs of the running configuration. This polling operation is for quick replacement of faulty computers without requiring programmers to change

coding. The polling operation is not in the demonstration program due to ongoing hardware changes by Data General Corporation.

3.1 MTI System Startup

The MCA is used to send programs to all the computers, as well as to transfer the data once the programs have begun execution. In the machine which will have the PIM function, the Real Time Disk Operating System (RDOS) is loaded via the channel start option. It then has the capability, via the Command Line Interpreter (CLI) instruction MCABOOT, to send the save files of the DIM, CAM, TRKIN, and display RECM programs to seven selected computers. Those computers must be started at location 376, where location 376 contains the instruction NIOS 7 (send a start pulse to MCA receive Channel 7) and where location 377 contains the instruction JMP 377 (jump in place). The sending computer, the PIM, will be given the following series of instructions:

```
MCABOOT MCAT:9 DDIMI2A.SV/S
MCABOOT MCAT:2 DCAM3RA.SV/S
MCABOOT MCAT:4 DCAM3RA.SV/S
MCABOOT MCAT:3 DCAM3RA.SV/S
MCABOOT MCAT:6 DCAM3RA.SV/S
MCABOOT MCAT:8 TRKIN15A.SV/S
MCABOOT MCAT:5 DRECM18A.SV/S
```

This will accomplish transmitting via the MCA bus the save file (filename.SV) named to the machine with the MCA address specified

(MCAT: address), provided that machine is listening (i.e., set up with the 2 instructions at locations 376 and 377). Once a machine has received its program from the PIM, control is passed to its starting address and execution begins.

3.2 MTI System Initialization

Once certain variables have been initialized in the DIM, CAM, TRKIN, and display RECM programs, they set up to receive via the MCA bus a 16 word header. Then all the machines remain in a ready state until the PIM sends a reset header and begins its data transfer. The program returns to this ready state whenever a machine has no other function than 1) to listen for any machine sending it a header or acknowledge block, or 2) to listen for a data block from a specific machine to which it is locked. By using the interrupt system, a program can be ready to accept a header, acknowledge or data block while processing data, thereby reducing the amount of time spent in a pure ready state to a minimal amount. This version of the program (demonstration version) uses this approach only in the track initiation computer (TRKIN) for some limited input types. It is expected that an operational version will make more extensive use of the interrupt system. Some of the more obvious changes have been noted elsewhere in this document.

The PIM machine begins its execution by the command DPIM25A (carrige return)[1] being typed on the console. The save file DPIM25A

[1] DPIM25A is the current filename of the PIM program. This could vary.

is then loaded into memory as RDOS is released. It first sends out reset headers to all machines except the display RECM, and then it begins to input data from the paper tape reader and send that data in unsorted form to the DIM and in sorted form to the CAMs. Processing continues as long as there is data being input to or cycling through the PIM.

3.3 Polling

Polling is a function which has been coded and will be added to the programs once the transmitter time out function is restored to each MCA board. Transmitter timeout is currently undergoing modification by Data General Corporation. Polling will enable all computers in the system, except for the controlling machine and any other machines with hardware limitations, to be loaded with either the DIM, CAM, TRKIN or display RECM programs. Each program will remain in a dormant ready state until the PIM sends it a polling reset header.

The PIM will start at the lowest available MCA address in the system and transmit a polling reset header to that machine at the same time setting itself up to receive a ready acknowledge block. Once it has successfully transmitted its polling reset header and received the acknowledge block, it will determine from the acknowledge block the type of function that machine's program has. The PIM then

will check its assignment list; if that function has already been assigned a machine address, it will send a negative reply to the machine with which it is currently communicating and then transmit a polling reset header to the machine with the next highest MCA address. However, if that function has not been assigned a machine, the PIM will assign it and then transmit a polling reset header to the machine with the next highest MCA address.

Once the assignment list is full, i.e. 1 DIM function, 4 CAM functions, 1 TRKIN function, and 1 display RECM function, the PIM sends assignment reset headers to each machine in the assignment list so that each may update the MCA addresses of the machines with which it needs to communicate. This polling function, therefore, allows all machines either to be operational or in a dormant ready state. Should a hardware difficulty occur with one machine whereby it is cut off from the MCA bus, the PIM machine can be instructed, via the setting of key 1 or by an internal time check, to send polling reset headers and then assignment reset headers to the remaining machines, thus transferring the function of the machine which failed to another machine.

4.0 FUNCTIONAL DESCRIPTION OF PROGRAMS AND SUBROUTINES

This section offers a detailed description of the function and usage of each program and subroutine used in the MTI processor. The following general note is applicable to the detailed descriptions of all main programs.

During the initialization/reset logic, following the clear MCA transmitter and clear MCA receiver instructions, tests are made as to whether the busy and done flags for the functions are immediately set to zero. If they are not, the program will halt. This indicates a probable hardware failure with the MCA, and it is recommended that diagnostics are run.

4.1 PIM

4.1.1 Function

The function is to take in data from the input device, currently the paper tape reader, sort it, and pass it to the 4 CAMS. The PIM also passes the unsorted input data to the DIM and the display RECM.

4.1.2 Usage

(a) Calling sequence

When RDOS is operational in the computer to be used for the PIM function, the name of the current PIM save file is typed on the console. This passes control from RDOS to the PIM program.

(b) Input

The input is currently a paper tape, where the order of the data is upper byte of X, lower byte of X, upper byte of amplitude and Y, lower byte of Y. Null bytes separate the frames of data. The beginning of a frame is sensed when 2 bytes with all bits set are read; the end of a frame is sensed when 10 null bytes in a row are read. Up to 5,000 X, Y pairs may be read into one frame. The data must be in order according to Y.

(c) Output

- o All of the unsorted input data is sent to the DIM and display RECM via subroutine SENDP.
- o All the coordinate pairs where Y<256 are sent to the 1st CAM via subroutine PCSRT.
- o All the coordinate pairs where $256 \leq Y < 512$ are sent to the 2nd CAM via subroutine PCSRT.
- o All the coordinate pairs where $512 \leq Y < 768$ are sent to the 3rd CAM via subroutine PCSRT.
- o All the coordinate pairs where $768 \leq Y < 1024$ are sent to the 4th CAM via subroutine PCSRT.

(d) Error messages - none

(e) Subroutines used:

- o SENDP - internal to PIM
- o PCSRT - internal to PIM

o TRNSV2 - external to PIM

o RCVSV1 - external to PIM

4.2 SENDP (Subroutine of PIM)

4.2.1 Function

SENDP creates the header block describing the current frame of PIM data, sends that header block to the machine whose MCA address is specified, and when that transmission is successful, sends the current frame of PIM data to that same machine.

4.2.2 Usage

(a) Calling sequence

LDA 2, MCA address

JSR SENDP

Accumulator 2 (AC2) must contain the MCA address of the machine to which the PIM data is to be sent.

(b) Input

The input is the MCA address in AC2 to which the header and data blocks are to be sent, the PIM data and the word count and frame # of the PIM data.

(c) Output

The subroutine transmits the header and data blocks to the requested machine, but does not alter any data or return any data or return any new values to the PIM.

(d) Error messages - none

(e) Subroutine used:

o TRNSV2 - external to PIM

4.3 PCSRT (Subroutine of PIM)

4.3.1 Function

The function is to sort data into a maximum of 4 groups, based on the Y value of each X, Y pair, and to send these 4 groups to the 4 CAM machines. Sorting is done by starting with the 256th word in the PIM data block (the Y value of the 128th X, Y pair) and comparing it to the bound value 256. If Y<256, advance another 256 locations and test again. If Y≥256, back up 2 locations and test again. Once Y becomes less than 256, send this section of the data to the first CAM. Then, setting a new base address to the (last location sent +1), continue the sort in the same manner, sending all the coordinate pairs where 256≤Y<512 to the second CAM, where 512≤ Y<768 to the third CAM, and where 768≤Y<1024 to the fourth CAM.

If the data is exhausted before all 4 CAMS have been accessed, PCSRT sends zero words to the CAMS for which there is no data.

4.3.2 Usage

(a) Calling sequence

LDA 1, Positive # of words to be sorted

LDA 2, Base address of PIM data

JSR PCSRT

(b) Input

The input is the positive word count of the PIM data, the base address of the PIM data, and the PIM data itself.

(c) Output

The subroutine transmits a header to each CAM describing the data to be sent and waits for a ready acknowledge block from the CAM. If the header is acceptable, it transmits the sorted data block. If the header is not acceptable, it transmits the header again.

(d) Error messages indicated by a halt:

o Ready Acknowledge From Wrong Machine

After the header describing the sorted data has been sent to a particular CAM, the subroutine listens for a ready acknowledge block. When it receives such a block, but it is from a different CAM than the one to which it just sent the header block, the program halts.

o Correct Machine - Wrong Type of Code

As described above, the PIM listens for a ready acknowledge block after it transmits a header to a CAM. When it receives the acknowledge block, if it is from the correct CAM but it is not a ready acknowledge block due to its type code, the program halts.

(e) Subroutines used:

o TRNSV2 - external to PIM

o RCVSV1 - external to PIM

4.4 TRNSV2 (General Subroutine)

4.4.1 Function

The function is to read in the present contents of the MCA transmitter address counter, word counter, and status, and to test for transmitter time out and transmitter count not done.

If the transmitter has timed out, the MCA transmitter is cleared and control is passed to the special return address so that the transmission may be repeated.

If the transmitter count is not done, TRNSV2 tests to see if the block just sent was a header, and if so, if only 5 words were transmitted. Under this condition, processing continues, as this indicates that the receiving machine was listening for a ready acknowledge block rather than a header.

4.4.2 Usage

(a) Calling sequence

LDA 1, zero if data or acknowledge block was just sent,

one if header block was just sent

LDA 2, address to which to return if retransmission is

necessary

JSR TRNSV

Note: The general calling name is the same as subroutine TRNSV1. The file names on the disc differ.

(b) Input

The input is the header or data/ready indicator passed via accumulator 1 and the special return address passed via accumulator 2.

(c) Output

No data values are affected, but the address to which control is passed is variable.

(d) Error messages indicated by a halt:

o Transmitter Count Not Done on Data or Acknowledge

When the transmitter count done bit is not set, and the indicator passed via accumulator 1 indicates that the transmission was of data or a ready block, then the program halts.

o Take Present Contents of Word Counter

Add to +11

If result = 0, then this was case where 5 words of a 16 word header were accepted

If result ≠ 0, then this is a legitimate transmit count not done

(e) Subroutines used - none

4.5 TRNSV1 (General Subroutine)

4.5.1 Function

TRNSV1's function is to read in the present contents of the MCA transmitter address counter, word counter, and status, and to test for transmitter time out and transmitter count not done.

If the transmitter has timed out, the MCA transmitter is cleared and control is passed to the special return address so that the transmitter may be repeated.

If the transmitter count is not done, the program halts.

4.5.2 Usage

(a) Calling sequence

LDA 2, address to which to return if retransmission is necessary

JSR TRNSV

Note: The general calling name is the same as subroutine TRNSV2. The file names on the disc differ.

(b) Input

The input is the special return address for retransmission passed via accumulator 2.

(c) Output

No data values are affected, but the address to which control is passed is variable.

(d) Error message indicated by a halt:

o XMTR Count Not Done

The transmitter did not send out all its words before the receiver stopped listening. The present contents of the MCA transmitter word counter shows the negative number of words still to be transmitted.

(e) Subroutines used - none

4.6 RCVSV1 (General Subroutine)

4.6.1 Function

RCVSV1's function is to read in the present contents of the MCA receiver address counter, word counter, and status, and to test for receiver time out and receiver count not done.

If the receiver has timed out or if the receiver count is not done, the program halts.

4.6.2 Usage

(a) Calling sequence

JSR RCVSV

Note: The general calling name is RCVSV. The file names on the disc have the prefix RCVSV1.

(b) Input - none

(c) Output

No data values are affected.

(d) Error messages indicated by a halt:

o RCVR Time Out

This indicates that a block transfer is in progress, but that no data has been received for 10 milliseconds. When the receiver time out bit is set, suspicious behavior is indicated, as it cannot be set by normal termination such as transmitter word count overflow.

o RCVR Count Not Done

The receiver did not receive all the words expected by its word counter. The present contents of the MCA receiver word counter show the negative number of words still to be received.

(e) Subroutines used - none

4.7 DIM

4.7.1 Function

The DIM's function is to receive data from the PIM, to sort it into a maximum of 4 groups, based on the Y value of each X, Y pair, and to send these 4 groups to the 4 CAM machines. Sorting is done by examining each X, Y pair in order, eliminating all X, Y pairs where Y's amplitude (bits 0-3) is less than the amplitude read in via the keys. The boundaries of Y's value for each CAM are described in c below. If the data is exhausted before all 4 CAMs have been accessed, the DIM sends zero words to the CAMs for which there is no data.

4.7.2 Usage

(a) Calling sequence

The DIM program is sent to the proper machine via an MCABOOT instruction from a machine in the system which has RDOS in an operational mode, usually the machine which later has the PIM function. Control then passes to the starting address of the DIM. Certain variables are initialized, and the DIM is set up to receive, via MCA, its first header. It waits at this point until its MCA receive done flag is set.

(b) Input

The input, received via the MCA bus from the PIM, is 1) a header block describing the data and 2) the data itself. Each frame contains data in X, Y order, where bits 0-3 of Y contain the amplitude. The data must be in order according to Y.

The input received via the keys on the computer is amplitude. It should appear in keys 8-11, with all other keys off. It is read once for each frame, and it is tested against each X, Y pair in the frame.

(c) Output

o All the coordinate pairs where the amplitude is greater than or equal to the value in the keys and where Y<256 are sent to the 1st CAM.

- o All the coordinate pairs where the amplitude is greater than or equal to the value in the keys and where $256 \leq Y < 512$ are sent to the 2nd CAM.
- o All the coordinate pairs where the amplitude is greater than or equal to the value in the keys and where $512 \leq Y < 768$ are sent to the 3rd CAM.
- o All the coordinate pairs where the amplitude is greater than or equal to the value in the keys and where $768 \leq Y < 1024$ are sent to the 4th CAM.

(d) Error messages indicated by a halt:

- o Not Data Code

  If PIM data was expected, but the 1st word in the block received was neither a header code nor a data code, the program will halt.

- o Odd # of Words

  The word count representing the amount of PIM data is odd. This is not allowable, as all data should be in X, Y two word groups.

- o Ready Acknowledge From Wrong Machine

  After sending out the header to a particular CAM, setting up to receive a ready acknowledge block, and having both those functions completed, it is found that the CAM to which the header was sent and the CAM which sent the ready block are not the same.

o Correct Machine - Wrong Type of Code

This is a continuation of the above situation, where the correct CAM sent a block to the DIM, but it was not a ready acknowledge type block.

(e) Subroutines used:

o TRNSV2 - external to DIM

o RCVSV1 - external to DIM

4.8 CAM

4.8.1 Function

There are 4 CAMS in the system, each receiving sorted data from the PIM and the DIM. Each CAM stores the DIM data in Table 2 and the PIM data in Table 1. The star cancellation subroutine establishes a bit matrix from the X, Y coordinate pairs in Table 1; and then, for each X, Y coordinate pair in Table 2, it establishes a bit pattern and determines if that pattern exists at the corresponding address of the bit matrix. If it does not, it stores that Table 2's X, Y coordinate pair in the leaker table. Once all Table 2 data has been exhausted, the CAM program sends a header describing the leakers and the leaker data itself to both the display reconstruction mini (RECM) and the track initiation machine (TRKIN).

4.8.2 Usage

(a) Calling sequence

The CAM Program is sent to the proper machines via an

MCABOOT instruction from a machine in the system which has RDOS in an operational mode, usually the machine which later has the PIM function. Control then passes to the starting address of the CAM. Certain variables are initialized, and the CAM is set up to receive, via MCA, its first header. It waits at this point until its MCA receive done flag is set.

(b) Input

The input received via the MCA bus is a header block describing the DIM data, the DIM data itself, a header block describing the PIM data, and the PIM data itself.

(c) Output

The output consists of the header describing the leaker data and the leaker data itself, which is sent via the MCA bus to the display RECM and TRKIN.

(d) Error messages indicated by a halt:

o Header Block Has Wrong Quadrant Indicator

The quadrant indicator in the eighth word of the DIM header block does not equal the MCA address of this CAM.

o Wrong Block Followed DIM Header

The block received into the DIM data area does not contain the proper data code.

o Quadrant Indicator is Wrong

Same as(d)1 but for PIM header

o Wrong Block Followed PIM Header

Same as (d)2 but for PIM header

o Not Header or Ready Block-Illegal

When the CAM program transmits a header block to the track initiation machine, it sets up a receive for a ready block. When the transmit and receive flags are both set, if the ready block contains neither a ready code nor a header code, the program halts.

o Ready Reply From Wrong Machine

When the condition described above exists, and the ready block is not from the track initiation machine, the program halts.

(e) Subroutines used:

o TRNSV1 - external to CAM

o RCVSV1 - external to CAM

o CNCLS - external to CAM

4.9 CNCLS (Subroutine of CAM)

4.9.1 Function

The star cancellation routine takes 2 data lists from the CAM, Table 1 which represents PIM data and Table 2 which represents DIM

data. It establishes a bit matrix from the X, Y coordinate pairs of Table 1 in the following manner: XYAD = base address of bit matrix + (bits 11-15 of Y)·(64) + bits 6-11 of X, shifted right 4 times. At locations XYAD, XYAD + 64, and XYAD + 128, it stores a bit pattern, based on the address of the bit pattern table + bits 12-15 of X. For each X, Y coordinate pair in Table 2, using the same method it establishes a bit pattern and determines if that pattern exists at the corresponding XYAD of the bit matrix. If it does not, it stores that Table 2's X, Y coordinate pair in the leaker table.

4.9.2 Usage

(a) Calling sequence

The CAM program will set the following values which are entry points of CNCLS:

- o TB1AD - the address of Table 1 data from the PIM
- o TB2AD - the address of Table 2 data from the DIM
- o CTTB1 - the negative word count of Table 1 data
- o CTTB2 - the negative word count of Table 2 data

The calling instruction is:

```
JSR CNCLS
```

(b) Input

The input is the Table 1 data, its negative word count, the Table 2 data, and its negative word count.

(c) Output

The output consists of the positive number of X, Y coordinate pairs of leakers (CTLKR), the leaker overflow variable (LKOVF), and the leaker data address (LK1).

(d) Error message indicated by a halt:

o Odd # of Words

This error halt may occur in 3 places, indicating that either Table 1 data or Table 2 data has an odd number of words according to its counter. Since data lists are expected to contain X, Y pairs for computation purposes, an odd number of words is an error.

(e) Subroutines called:

The only subroutines called are those internal to the routine which establish and store the bit patterns in the bit matrix.

4.10 TRKIN

4.10.1 Function

The TRKIN program is set up to receive leaker data from all 4 CAMS. When it has received leaker data from all 4 CAMS, i.e., a frame, it listens for leakers from all 4 CAMS again. When it has received 3 frames, it uses relative frame 2 as the basis and determines if the average of a given set of coordinate pairs from relative frames 1 and 3 matches a coordinate pair in relative frame 2, thus establishing a track.

When such a match occurs, the coordinate pairs from frame 1 and frame 3 are stored in a track list, and after processing all points for a given 3 frames, this list is transmitted to the display Reconstruction Mini (display RECM).

As a new frame is read in, the third frame is dropped, so that the new frame becomes frame 1, frame 1 becomes frame 2, and frame 2 becomes frame 3. The tracking algorithm is repeated with the addition of each frame.

4.10.2 Usage

(a) Calling sequence

The TRKIN program is sent to the proper machine via an MCABOOT instruction from a machine in the system which has RDOS in an operational mode, usually the machine which later has the PIM function. Control then passes to the starting address of TRKIN. Certain variables are initialized, and TRKIN is set up to receive, via the MCA bus, its first header. It waits at this point until its MCA receive done flag is set.

(b) Input

The input received via the MCA bus consists of header blocks describing the leaker data and the actual leaker data from the 4 CAMS. The track frame filling buffer is considered full when a header block and a data block of

leakers have been received from each of the 4 CAMS. TRKIN will not accept another header from a CAM until all 4 CAMS have sent their data and the filling buffer has become frame 1.

(c) Output

The output to the display reconstruction mini (display RECM) consists of a header describing the track initiation data, the track initiation data itself which is composed of X, Y coordinate pairs from frames 1 and 3, a header describing frame 2 data, and the entire block of frame 2 data itself.

(d) Error messages indicated by a halt:

- o This Is Not A Header

  The data just received in the reset header block is not a header, as indicated by its lack of a reset header code.

- o This Is Not A Reset Header

  Although the block just received in the reset header area is a header type, it is not a reset header.

(e) Subroutines called:

- o INPTF - internal to TRKIN
- o BUMP - internal to TRKIN
- o INTSV - internal to TRKIN

- o SQSHF - internal to TRKIN
- o SETHD - internal to TRKIN
- o SETM - internal to TRKIN
- o CHCKM - internal to TRKIN
- o LSXYZ - external to TRKIN
- o TRNSV1 - external to TRKIN
- o PCVSV1 - external to TRKIN

4.11 INPTF (Subroutine of TRKIN)

4.11.1 Function

INPTF establishes the variables for the initialization of the filling buffer and sets up the MCA receive for the first header from any one of the 4 CAMS.

4.11.2 Usage

(a) Calling sequence

```
JSP  INPTF
```

(b) Input

None

NOTE: Throughout the description of TRKIN and its subroutines, the filling buffer refers to the F area in the code; frame 1 refers to the X area in the code; frame 2 refers to the Y area in the code; and frame 3 refers to the Z area in the code.

(c) Output

The CAM working list is established from the CAM acceptable list; the full indicator is cleared; the CAM counter is set to -4; the current header the current data pointers are established; and the MCA receiver is unlocked and the receiver turned on to listen for a header.

(d) Error messages - none

(e) Subroutines called - none

4.12 BUMP (Subroutine of TRKIN)

4.12.1 Function

BUMP clears the full indicator and rotates the data pointers, header pointers, and buffer counters when the filling buffer becomes full.

4.12.2 Usage

(a) Calling sequence

JSR BUMP

(b) Input - none

(c) Output

The variables named in 4.12.1 are updated.

(d) Error messages - none

(e) Subroutines called - none

4.13 INTSV (Subroutine of TPKIN)

4.13.1 Function

INTSV services the MCA receive interrupts. Header input from CAMS is done on an interrupt basis; the data input which follows the header is done within INTSV using flag tests rather than interrupts. All other MCA transmissions and receives done in TPKIN or its subroutines are done using flag tests.

INTSV reads in the present contents of the receiver word address, the present contents of the receiver word counter, and the receiver status. After checking the status bits, it can follow one of three paths. 1) If the header block is a reset header, it sets the reset indicator, restores the accumulators, and returns to the location where the interrupt occurred. 2) If the header comes from a CAM in the CAM working list, an acceptable ready acknowledge is sent to the CAM and the leaker data is received. That CAM is removed from the working list. If there are more CAMS to send data for this frame, it unlocks the MCA receiver and sets up to receive another header. It then restores the accumulators, enables interrupts, and returns. If it has received headers and data from all the CAMS, then it sets the full accumulator, restores the accumulators and returns. 3) If the header does not come from a CAM in the working list, it sends back a not acceptable ready block to the machine which sent the header, unlocks the MCA receiver, and sets up to

receive another header. It restores the accumulators, enables interrupts, and returns.

4.13.2 Usage

(a) Calling sequence

During program initialization location 0 on the zero page is cleared and location 1 on the zero page is set to the address of subroutine INTSV. Whenever interrupts are enabled and an interrupt occurs, the present contents of the program counter is stored at location 0 on the zero page and control passes to the address in location 1 on the zero page.

(b) Input

The input consists of the contents of the current header block which triggers the interrupt.

(c) Output

Based on the contents of the current header block, the output could be a full header and data filling buffer, another addition to the header and data filling buffer, or nothing.

(d) Error messages indicated by a halt:

o Is Interrupt On MCA Receive? No

This error occurs when the interrupt acknowledge shows the interrupt to be from some channel other than MCA receive.

o Receiver Time Out

This indicates that a block transfer is in progress, but that no data has been received for 10 milliseconds. When the receiver time out bit is set, suspicious behavior is indicated, as it cannot be set by normal termination, such as transmitter word count overflow.

o Receiver Count Not Done

The receiver did not receive all the words expected by its word counter. The present contents of the MCA receiver word counter shows the negative number of words still to be received.

o Is This A Header Block? No

The only type of block which should have caused an interrupt is a header block.

o Block Just Received Was Not Data Block

After the MCA bus received a block of data which was expected to be a data block, its type code is not a data code.

(e) Subroutines called:

o TRNSV1 - external to INTSV and TRKIN

o RCVSV1 - external to INTSV and TRKIN

4.14 SQSHF (Subroutine of TRKIN)

4.14.1 Function

SQSHF squashes in place the filling buffer's data from 4 CAMS into 1 list, eliminating unused locations and extra 2 word data code blocks. It also establishes the value of CTRF, which is the total negative word count of the list.

4.14.2 Usage

(a) Calling sequence

JSR SQSHF

(b) Input

The input to SQSHF is a full data filling buffer and a zero value in CTRF.

(c) Output

A compressed data filling buffer and the variable CTRF containing the negative word count of the data is the output.

(d) Error messages indicated by a halt:

o # of Data Words Sensed In F Doesn't Agree With Sum of Header Counts

This halt checks the word count in the data filling buffer by comparing a word by word count with the sum of the 4 word counts taken from the header filling buffer.

(e) Subroutines called - none

4.15 SETHD (Subroutine of TRKIN)

4.15.1 Function

SETHD sets up the header describing the current frames 1, 2, and 3 which will be sent to the display Reconstruction Mini (display RECM).

4.15.2 Usage

(a) Calling sequence

JSR SETHD

(b) Input

SETHD uses values contained in the header blocks describing all headers for frame 1 and for frame 3. It also uses the variable CTRY, which is the negative word count of data in frame 2.

(c) Output

The output is a 16 word header block which describes the track data and frame 2 data which is to be sent to the display RECM. An outline of this header's contents is contained in Appendix III.

(d) Error messages - none

(e) Subroutines called - none

4.16 SETM (Subroutine of TRKIN)

4.16.1 Function

SETM sets up a bit matrix of 256 by 16 words from data in

relative frame 1. By the time that this bit matrix is used, relative frame 1 will be relative frame 2.

Each X, Y coordinate pair in relative frame 1 is used to establish a bit pattern in the bit matrix according to the following algorithm:

YAD = Bits 4-11 of (Y*4) + base address of bit matrix

BITX = Bits 12-15 of (X÷4)

BTPAT = Contents of (base address of bit pattern table + BITX)

XYAD = YAD + [Bits 8-11 of (X÷4)]÷16

Then SETM inclusive ors the contents of XYAD with BTPAT and stores the result at XYAD.

4.16.2 Usage

(a) Calling sequence

JSR SETM

(b) Input

SETM's input is relative frame 1 data and its negative word counter.

(c) Output

SETM's output is a 256 by 16 word bit matrix, established according to the data in relative frame 1.

(d) Error message indicated by a halt:

o Odd Counter For X Area Data

Since all data is expected to be in X, Y coordinate pairs, an odd number of words indicates an error.

(e) Subroutines called - none

4.17 CHCKM (Subroutine of TRKIN)

4.17.1 Function

CHCKM detects potential tracks by taking the average of each X, Y coordinate pair in relative frame 1 with every X, Y coordinate pair in relative frame 3; and, after computing the appropriate bit matrix pattern and offset for this average coordinate pair, checks that bit matrix address for that pattern. If a match exists, the X, Y coordinate pair from relative frame 1 and the X, Y coordinate pair from relative frame 3 are stored in the track list.

The following algorithm is used:

SUMY = (Y of frame 1 + Y of frame 3) *2

MTXCT = Bits 4-11 of SUMY + base address of bit matrix

SUMX = (X of frame 1 + X of frame 3) ÷8

BITX = Bits 12-15 of SUMX

BTPAT = contents of (base address of bit pattern table + BITX)

XYAD = MTXCT + (Bits 8-11 of SUMX) ÷16

Then CHCKM logical ands the contents of XYAD with BTPAT. If this result is nonzero, the coordinate pairs are stored in the track list as described in the above paragraph.

4.17.2 Usage

(a) Calling sequence

```
JSR CHCKM
```

(b) Input

CHCKM's input consists of relative frame 1 data, relative frame 3 data, the negative word counters of both frames, and a 256 by 16 word bit matrix, representing the data in relative frame 2.

(c) Output

CHCKM's output is a list of potential track X, Y coordinate pairs from relative frames 1 and 3 and a negative word counter of the list's contents.

(d) Error messages indicated by a halt:

o TICTR Overflow

The variable TICTR, which contains the negative word count of the data in the track list, has exceeded - 1024, the current limit to the track list.

o Odd Counter For Z Area Data

Since all data is expected to be in X, Y coordinate pairs, an odd number of words indicates an error.

o Odd Counter For X Area Data

See explanation above

(e) Subroutines called - none

4.18 LSXYZ (Subroutine of TRKIN)

4.18.1 Function

When key 0 is set, program TPKIN calls LSXYZ to print out on the teletype the track data detected in subroutine CHCKM and the accompanying relative frame 2 data. If no track data was found for the 3 current frames, that message is printed.

4.18.2 Usage

(a) Calling sequence

The keys are read by TPKIN, and if key 0 is set, JSP LSXYZ

(b) Input

TRKIN passes to LSXYZ via zero page locations 50-53, the address of the track data, the address of relative frame 2 data, the negative word counter of the track data, and the negative word counter of the relative frame 2 data.

(c) Output

The output is a teletype listing of the octal values of the track data and relative frame 2 data.

(d) Error messages indicated by a halt:

o If Count Is Not Zero, It Must Be Negative

This halt indicates the track data counter is in error.

o This Indicates An Odd Y Count, Which Is Wrong

Since all data is expected to be in X, Y coordinate pairs, an odd number of words indicates an error.

(e) Subroutines used:

o LNOUT - internal to LSXYZ

This subroutine outputs a line of 58 ASCII characters to the teletype.

o PCONV - internal to LSXYZ

This subroutine converts a 16 bit word to octal ASCII code.

4.19 Display RECM

4.19.1 Function

The display RECM program continuously receives header and data blocks from the PIM, the 4 CAMS, and TRKIN; and it continuously displays, depending on the key settings, the data sent by the PIM representing the current input frame, the leakers sent from the CAMs representing the cancellation of the current frame against the previous frame, or the track data sent by TRKIN.

In order to service done flags from both the MCA receive and the display, a loop named FLAGS serves as the main reference in the program. It checks for the done flag on the display to be set, and once it is, it checks for the done flag on the MCA receive. If it is not set, control is passed to REDIS, where it is determined, based on counters and the keys, what will be displayed next.

The keys' functions and priorities are as follows. If either key 0 or key 1 is set, only track data is displayed. Key 0 controls the display of relative frame 1, or X area, track data. Key 1 controls the display of relative frame 3, or Z area, track data. Each set of track data for which a key is set is displayed 50 times,

first frame 1 data, then frame 3 data, before the keys are read again.

If neither key 0 nor key 1 is set, keys 2-8 are taken to be the number of times the PIM data will be displayed before control is passed to display the leaker data from the 4 CAMs the number of times indicated by keys 9-15. Once this sequence is completed, the keys are read again. If no keys are set, the PIM data is displayed once, and then the keys are read again.

The MCA receive service area keeps track of whether a header block or data block is expected next, and if a data block is expected, the area knows which machine should be sending that block. Once a header is accepted from a machine the display PECM remains locked to that machine until the data block is received. It unlocks its receiver only when listening for another header.

4.19.2 Usage

(a) Calling sequence

The display RECM program is sent to the proper machine via an MCABOOT instruction from a machine in the system which has RDOS in an operational mode, usually the machine which later has the PIM function. Control then passes to the starting address of the display RECM. Certain variables are initialized, and the display RECM is set up to receive, via MCA, its first header. It displays one zero point until it receives data.

(b) Input

The input received via the MCA bus consists of header blocks and data blocks of a whole frame's data from the PIM, leaker data from the 4 CAMs, and track data from TPKIN.

The input received via the keys directs the program and establishes counters.

(c) Output

The output is the X, Y display of track data, or frame data and/or leaker data.

(d) Error messages indicated by a halt:

o Count should always be even

This refers to the count of the PIM data. Since all data is expected to be in X, Y coordinate pairs, an odd number of words indicates an error.

o RCVR Time Out

This indicates that a block transfer is in progress, but that no data has been received for 10 milliseconds. When the receiver time out bit is set, suspicious behavior is indicated, as it cannot be set by normal termination, such as transmitter word count overflow.

o RCVR Count Not Done

The receiver did not receive all the words expected by its word counter. The present contents of the MCA receiver word counter shows the negative number of words still to be received.

- o Invalid MCA Code

  The machine which sent the current header block is not in the system.

(e) Subroutines Used:

- o MERGL - internal to dispaly RECM

  MERGL masks off either one or two low bits from the 10 bit X and Y words and merges them into one 16-bit word where bits 0-7 = Y and 8-15 = X. MERGL merges the leaker data from the 4 CAMs.

- o MRGTI - internal to display RECM

  MRGTI merges either the relative frame 1 section of the track data or the relative frame 3 section of the track data, as described in MERGL above.

5.0 MTI DEMONSTRATION SOFTWARE OPERATING CHARACTERISTICS

This section describes the testing which has been accomplished for the MTI software. Since this program does not have access to real time inputs, the testing has been very limited. Basically, a rough estimate of the actual runtime cycle has been obtained. The programs (and machines) have been life tested to determine any problems which might occur for long operating periods. Finally the detail of each process used to obtain numerical results has been verified.

5.1 Reliability Test

To test reliability of the MTI processor it was necessary to simulate some real time input, since no live inputs were available.

The PIM program was modified to ignore the necessary input from the preprocessor. In its place, a paper tape containing a simulated star field containing 5000 stars was loaded into the preprocessor data input area. The PIM program was then allowed to go through its natural MTI cycle by-passing only the replenishing of that data area. The effect was to have the PIM "see" the same FOV repeatedly. The remainder of the MTI process then proceeded as it normally would, except of course that cancellation was perfect.

Because of the rigid data flow structure required for control, however, data blocks were passing through all of the normal paths for every MTI cycle. For example data from the CAMS consisted of just

the header and the null data block. However, this test was representative for the major blocks of about 10,000 words each. Since this program was a test version, there was an additional large data transfer from the PIM to the display RECM.

Every known abnormal data transfer outcome was blocked by program halts. Any one program halt will stop the entire MTI complex because of the computer complex control scheme (Section 2).

The system was allowed to cycle in this manner for 200 hours before other requirements brought the test to a conclusion. There were no failures. The test MTI programs passed through an estimated 2.5 million cycles. Considering that the usable nighttime hours at a GEODSS sensor location will be on the order of 10, then one could expect that soft failure would occur less often than twice a month. The recovery from a soft error usually requires 2 minutes or less for the software in its present form. This time can be shortened with minor modifications to cause the loss of less than 10 seconds.

Hard failures in the multicomputer complex are of two basic types. The first and easiest type to deal with is the failure of a minicomputer. With the bus system, the faulty machine may be replaced while the software avoids using that machine. This is possible using the present software and hardware to a limited extent. Lack of conformity and the existence of some critical components such as the PIM and DIM computers which are the only computers connected to the disc currently pose limits to this flexibility.

5.2 Timing Test

Using the scheme described for reliability, a number of system cycles were visually counted within one minute. Approximately 120 total cycles were completed to give a 1/4 second cycle time. Sub portions of the MTI process are faster. The slowest portion of the program is cancellation which requires approximately .15 to .2 seconds for 10,000 stars. Much of the delay experienced for this test is due to the display technique used and the extra data distributed solely for display.

5.3 Coding Integrity

Each program operation was manually verified with abbreviated data sets taken at John Bryant State Park during the previous fiscal year. This data was recorded with an early version of a MITRE digital preprocessor. It contains a SIT tube with appropriate sampling circuitry. The FOV was ~1 degree and resolution was 525 x 525 cells. The frames were taken at a rate of 30 per second.

One series of four frames which contained the satellite PAGEOS passing through a star field were reduced with the MTI software and also by hand. At each stage of the process, namely sorting, cancellation, and track initiation, all the correct answers were laboriously computed by hand from a printout of the recorded data. These answers were verified against lists generated by the MTI software. (Temporary printout programs were installed along the MTI

process chain to obtain intermediate results.) Every data point was accounted for. The program behaved exactly as requested. Tracks were produced for PAGEOS. Because the data was neither centroided nor reduced to one equivalent point for each point cluster resulting from bright objects, a myriad of tracks occurred at the PAGEOS coordinates since each point pairing in the end point frames found a counterpart in the middle frame.

6.0 RECOMMENDATIONS

Recommendations resulting from this programming work are in two classes: 1) Changes are required to convert the demonstration program into a usable operational MTI device. 2) Should this work result in an operational device, there are several changes which should be made to any "copies" built from scratch.

6.1 Recommended Changes to Realize an Operational MTI Device

6.1.1 PIM

The PIM program logic to enable the input of real time data from the preprocessor has been coded but is not presently activated. That is, code relating to the preprocessor has been reduced to "comment" form in assembly language. Notations to this effect are present in the program listings. The paper tape input of data, and dummy cycling instructions to enable reliability testing should be removed.

The PIM program currently sends the entire data set for each FOV to the display RECM for each MTI cycle. This feature, although very useful for monitoring purposes, is not necessary and should be removed since it slows down MTI cycling.

In operational MTI software the PIM will have to file data frames on the fixed head disc and inform the DIM of where each frame is filed. It is envisioned that such filing would best be done by a scheme based on telescope pointing direction. The transmission of filing information to the DIM already exists in skeleton form; that is,

the file coordinates are not given. The header that precedes the data in the current system serves this purpose. The data block which is sent to the DIM will of course be eliminated when the disc filing scheme is installed. The telescope position will have to be obtained either from the site computer or directly from the mount at the GEODSS Experimental Test Site. Presumably these coordinates will be made to match the exposure to the FOV.

The PIM will also require bright star information from the site computer. It is expected that frame coordinates of 5 bright stars for each frame will be needed. These are for use in registration from frame to frame. The code for these functions does not exist.

6.1.2 DIM

The DIM coding to keep the dossier of frames placed on the disc by the PIM and the logic to effect a retrieval of that data does not exist. The scheme is envisioned as follows: The "delayed" frame will have to be retrieved from the disc by the DIM according to the header description received from the PIM regarding the current frame. The PIM will write the current frame onto a "Travelling Frame Buffer". The Travelling Frame is achieved by allocating space on the disc for one frame in addition to that number needed to achieve the MTI scan pattern. The DIM will know which frames have old data from a cross referenced list of frame file identifications and disc file pointers made up from headers received from the PIM. The PIM storage and DIM retrieval disc areas will never overlap using this method.

6.1.3 CAMS

The CAM program may stand as is until registration is required. The cancellation process can be used to remove misregistration of one frame with respect to another. Assuming that the only significant registration problems are translations, one can compute displacement in X and Y and adjust the computed sparse matrix positions for each point by the same amount. The sparse matrix is described in the section on cancellation.

6.1.4 TRKIN

The track initiation outputs should be sent to the site computer for checking against RSO's in the vicinity or for designation of a track to gather more information.

6.1.5 General

All of these programs are scattered with real time traps due to the demonstration of feasibility nature of this effort. These traps should be replaced by either error messages or a recovery to the reset mode. The recovery to a reset simply means that halts indicative of illogical system behavior would necessarily require a system restart. The natural program waiting place for a system restart is to clear all indications implying that the program has ever been used and then to wait for a new reset header. This implies that a new poll will be taken and some other machine will probably be removed from the system.

6.2 Recommended Alternative Approaches

6.2.1 Hardware

This hardware system was acquired in two phases. First, enough hardware was obtained to perform a minimal MTI role. Software development hardware such as a line printer, magnetic tape recorder, etc. was acquired later. One lesson became evident while the installation of the second shipment was in progress. It is extremely desirable to have complete hardware modularity of critical items such as computers. That is, the system should be made invulnerable to the failure of any one computer. Except for those computers with disc interface or MTI system I/O devices, that is currently true. The expenditure of additional funds to upgrade those elements without disc connections and MTI system I/O or to add more core at this time is clearly not necessary for this particular system due to its proven performance. However, in the future, field installations would benefit from considerations to achieve complete computing element interchangeability. It should be noted that once this interchangeability is achieved, software may be used to rearrange the element roles without physical intervention.

A further hardware design change which seems necessary, perhaps even for this system, would be the development of a simple technique for physically removing computers from the MCA bus and leaving the system operational.

Finally the hardware interconnection scheme, that is, push on connectors, should be replaced by a positive holding scheme. Connectors should be replaced by those which can be physically locked. The MCA bus cable length should also be shortened in order to achieve a 500 kHz word transfer bandwidth.

6.2.2 Software

The acknowledge block is currently of a different block length than the header block. These should be made to be identical length. This minor adjustment will simplify some of the program logic.

7.0 SUMMARY AND CONCLUSIONS

In summary, a commercially available collection of interconnected minicomputers was obtained for use as a parallel processor. By using software, the system of computers was configured to serve as an MTI processor with the potential of operation in real time. The snapshot MTI algorithms required to detect satellite motion in a field of up to 5000 stars was implemented. The implementation consists of frame-to-frame star cancellation, and a cancelled-frame to cancelled-frame target vector computation which eliminates single-frame false alarms. The scheme has a real-time cycle of approximately 1/4 to 1/3 second. This software is designed to be expandable for larger star fields and for adaptation to use in real time.

This work represented a very modest programming effort. That is, active programming development including design actually took two individuals less than 6 months. This was possible mainly through the modularity of this computer complex and the simplicity of inter computer conversation. In addition, this system has proven extremely versatile for expansion and alteration. For example, the programs were developed one at a time and their development sequence followed the path of the data through the MTI processor. As each new subprocess was added, a new computer or bank of computers came into play. The interactions of programs at the various stages were minimal due to the coordination scheme employed. This implies that the multi-

computer complex and basic software can become a flexible tool for any further development effort. Any technique for MTI which relies on a series of cancelled star fields for its data base can be programmed for the single computer which receives the appropriate output. For example, the sequence of cancelled frames from which track vectors are computed could instead be combined into one composite frame for the testing of "streak" detection algorithms.

The matrix method of list processing used for this software also seems to have general utility in analyses wherein two pictures which have been reduced to lists are to be compared. This advantage presumably will only apply for pictures which are 90% blank but have lists of more than 10 or so entries.

The results of the multiprocessor aspect of this work indicate that the coordination of several computers to operate simultaneously in a real-time environment is not only feasible but enjoys several advantages over the single computer with massive capabilities.

The risk involved in estimating the computer power needed to achieve any MTI scheme is greatly reduced. For example, should one find insufficient computing capacity for the cancellation process, the field of view may be further subdivided and the job distributed to a few more inexpensive computers to overcome the shortcoming. An underestimation of the requirements for a single larger machine, however, would mean that the entire machine would have to be

replaced. If one overestimated the requirements, small machines could be removed, or reallocated to other roles.

The results obtained with the MTI program itself point towards a high confidence in the computing ability of the Multi-minicomputer MTI system and its ability to handle real time problems at the GEODSS Experimental Test Site. Furthermore, there is considerable flexibility available to adapt the MTI software to whatever unforeseen qualities real data might possess. The question which remains is not whether the snapshot MTI approach is feasible but how well it will function in the vagaries of the real world environment.

APPENDIX I

PROGRAM LISTINGS

```
                                DPIM25A
  0001  DPIM
01                      .TITL DPIM
02                        .EXTN .UCEX
03                        .EXTN TRNSV,RCVSV
04                        .ENT DPIM
05                    ;   DEMONSTRATION VERSION OF PIM PROGRAM, WHICH
06                    ;   RECEIVES DATA FROM THE PREPROCESSOR AND SENDS
07                    ;  IT UNSORTED TO THE DIM R RECM, AND SORTED, TO THE 4 CAMS.
08                    ;   KEY OPTIONS --
09                    ;     IF KEY 0 IS SET, DO NOT SEND START TO PRE-
10                    ;     PROCESSOR UNTIL KEY 0 IS UNSET.  THIS WILL
11                    ;     ALLOW SLOWER PROCESSING OF DATA AND GIVE TIME TO TAKE
12                    ;     PICTURES OF DISPLAY OUTPUT.
13                    ;     IF KEY 1 IS SET, SEND RESETS TO DIM R CAMS R TRKIN
14                    ;     AND RESTART PIM PROGRAM.
15                    ;      IF KEY 13 IS SET, A LARGE TAPE IS
16                    ;      BEING READ IN.  WAIT UNTIL TAPE IS CHANGED.
17                    ;      IF KEY 14 IS SET, BYPASS THE READING
18                    ;      OF THE NEXT FRAME R USE THE CURRENT
19                    ;      FRAME FOR PROCESSING
20                    ;     IF KEY 15 IS SET, EXIT THE PIM PROGRAM, REMOVE
21                    ;     THE USER CLOCK AND WAIT FOR CONSOLE COMMAND
22                    ;     TO RETURN TO RDOS.
23                        .ZREL
24          000026        PREO=26   ; MNEMONIC FOR PREPROCESSOR
25 00000-000016'      .UCLOCK:UCLOCK
26 00001-000012 S10: 10.
27 00002-000000 SAVE: 0
28 00003-000000 LOC0: 0
29 00004-000000 LOC1: 0
30 00005-177773 M5: -5.
31 00006-177772 M6:-6.
32 00007-000000 CTR: 0
33 00010-154340 MMAX: -10000.
34 00011-000000 .PIM:0
35 00012-000000 PIMCT:0
36 00013-177760 M16: -16.
37 00014-000000 NONUL: 0
38 00015-000000   CT21:0
39 00016-154353 MMAX1: -10005.
40 00017-000000 URL04: 0
41 00020-000000 SAVEP:0
42 00021-000023-MCAAD:DMMCA
43 00022-000000 MCAKP:0
44 00023-110000 DMMCA:110000
45 00024-020000 C3MCA:020000
46 00025-040000 C4MCA:040000
47 00026-030000 C5MCA:030000
48 00027-060000 C6MCA:060000
49 00030-100000 T1MCA:100000
50 00031-050000 P1MCA:050000
51 00032-120000 PMMCA:120000
52 00033-000102'.PSTRT:RSTRT
53 00034-000637'.HEADP:HEADP
54 00035-000327'.TR1:TP1
55 00036-000342'.TR2:TP2
56 00037-000000 FRMCT:0
57 00040-000000 SCNCT:0
58 00041-000200'.PIMIN:PIMIN
59 00042-177777 .TRNSV:TRNSV
```

```
  0002  DPIM
01 00043-000314'.SENDP:SENDP
02 00044-000361'.PCSRT:PCSRT
03 00045-100004 RDCDE:-32764.
04 00046-000000 SVMCA:0
05 00047-100003 RDCDE:-32765.
06 00050-000000 .DTCDE:0
07 00051-177777 .RCVSV:RCVSV
08                ;   VARIABLES FOR SORT ROUTINE
09 00052-000000 SVRTN:0
10 00053-000000 OTLAD:0
11 00054-000024-CAMAD:C3MCA
12 00055-000000 CAMCT:0
13 00056-000074-RNDBS:ROUND
14 00057-000000 RNDPR:0
15 00060-000377 TSS:255.
16 00061-000400 TS6:256.
17 00062-177774 M4:-4.
18 00063-000000 CTRI:0
19 00064-000000 FINL:0
20 00065-177776 M2:-2.
21 00066-000000 RDCNT:0
22 00067-000000 FINLY:0
23 00070-007777 YMASK:7777
24 00071-000000 SAVE3:0
25 00072-177750 M24:-24.
26 00073-000010 PB:8.
27        000010 CAML=10
28        000020 CAMH=20
29        000030 NOCAM=30
30 00074-000400 BOUND:256.
31 00075-001000        512.
32 00076-001400        768.
33 00077-002000        1024.
34 00100-000000        0
35 00101-000000        0
36 00102-000000        0
37 00103-000000        0
38 00104-000000        0    ; STORAGE FOR CAM BASE ADDRESS- CAML
39 00105-000000        0
40 00106-000000        0
41 00107-000000        0
42 00110-000000        0
43 00111-000000        0
44 00112-000000        0
45 00113-000000        0
46 00114-000000        0    ; STORAGE FOR CAM FINAL ADDRESS - CAMH
47 00115-000000        0
48 00116-000000        0
49 00117-000000        0
50 00120-000000        0
51 00121-000000        0
52 00122-000000        0
53 00123-000000        0
54 00124-000000        0    ; STORAGE FOR CAM WORD COUNT - NOCAM
55 00125-000000        0
56 00126-000000        0
57 00127-000000        0
58 00130-000000        0
59 00131-000000        0
```

```
  0003  DPIM
01 00132-000000          0
02 00133-000000          0
03 00134-000467'.TRC1:TRC1
04 00135-000526'.TRC2:TRC2
05 00136-000000 K:0
06 00137-000632'.READY:READY
07 00140-000000 SAVR4:0   ; ADDRESS IN RIM ARRAY WHERE
08                        ; DTCDE WORD WAS STORED
09 00141-000000 SAVW0:0   ; 2 RIM WORDS REPLACED BY
10 00142-000000           0 ;  -32767. & 0
11 00143-000000 FIXUP:0   ; IF FIXUP = 0, NO RIM WORDS WERE
12                        ; REPLACED BY DTCDE WORDS
13                        ; IF FIXUP=1, 2 RIM WORDS WERE
14                        ; REPLACED BY DTCDE WORDS
15 00144-011610 RS000:5000.
16 00145-146170 M5000:-5000.
17 00146-000621'.NMORE:NMORE
18 00147-000411'.NXTI:NEXTI
19 00150-000426'.SHRT:SHORT
20 00151-000400 .NMAX: 400
21 00152-100001 DTCDE:-32767.
22 00153-000437'.GTIT:GOTIT
23                        .NREL
24 00000'020001-DRIM: LDA 0,S10
25 00001'024000-      LDA 1,.UCLOCK
26 00002'006017       .SYST
27 00003'021001       .RUCLK
28 00004'004006       JSR ERR
29 00005'000400       JMP .
30 00006'006017       .SYST
31 00007'021002       .RUCLK
32 00010'004002       JSR ERR
33 00011'000400       JMP .
34 00012'054002-ERR:  STA 3,SAVE
35 00013'006017       .SYST
36 00014'006400       .ERTN
37 00015'002002-      JMP @SAVE
38 00016'054002-UCLOCK:STA 3,SAVE
39 00017'060277       NIOC CPU    ; DISABLE INTERRUPTS
40               ;    FIND OUT NMAX ADDRESS AND STORE -32767. & 0
41               ;    START  THERE, SET .DTCDE & .PIM ADDRESSES.
42 00020'034151-      LDA 3,.NMAX
43 00021'021400       LDA 0,0,3
44 00022'111000       MOV 0,2
45 00023'020152-      LDA 0,DTCDE
46 00024'041000       STA 0,0,2
47 00025'050050-      STA 2,.DTCDE
48 00026'151400       INC 2,2
49 00027'102400       SUB 0,0
50 00030'041000       STA 0,0,2
51 00031'151400       INC 2,2
52 00032'050011-      STA 2,.RIM
53 00033'000401       JMP .+1
54               ;    CONTROL PASSES HERE THE FIRST TIME, AND WHEN
55               ;    KEY 1 IS SET IS DETECTED DURING READING OF KEYS
56 00034'060207 RESET:NIOC MCAR
57 00035'063507       SKRBZ MCAR
58 00036'000400       JMP .
59 00037'063707       SKPDZ MCAR
```

```
  0004  DPIM
01 00040'000400          JMP .
02 00041'060206          NIOC MCAT
03 00042'063506          SKPBZ MCAT
04 00043'000400          JMP .
05 00044'063706          SKPDZ MCAT
06 00045'000400          JMP .
07 00046'060212          NIOC PIR    ; THIS WILL BE REMOVED WHEN PRE-
08                                   ; PROCESSOR INPUT IS USED
09 00047'102400          SUB 0,0
10 00050'040037-         STA 0,FRMCT
11 00051'040040-         STA 0,SCNCT
12 00052'034034-         LDA 3,.MEADR
13 00053'020045-         LDA 0,MDCDE
14 00054'041400          STA 0,0,3
15 00055'102000          ADC 0,0     ; -1
16 00056'041401          STA 0,1,3
17 00057'102400          SUB 0,0
18 00060'041402          STA 0,2,3
19 00061'041403          STA 0,3,3
20 00062'041404          STA 0,4,3
21 00063'041405          STA 0,5,3
22 00064'041406          STA 0,6,3
23 00065'041407          STA 0,7,3
24 00066'041410          STA 0,10,3
25 00067'041411          STA 0,11,3
26 00070'041412          STA 0,12,3
27 00071'041413          STA 0,13,3
28 00072'041414          STA 0,14,3
29 00073'041415          STA 0,15,3
30 00074'041416          STA 0,16,3
31 00075'041417          STA 0,17,3
32                  ;    SEND RESET HEADER TO DIM & 4 CAMS & TRKIN
33 00076'020006-         LDA 0,M6
34 00077'040007-         STA 0,CTR
35 00100'034021-         LDA 3,MCAAD
36 00101'000403          JMP RSRT1
37 00102'034022-RSTRT:LDA 3,MCAKP
38 00103'020013-RSRT1:LDA 0,M16
39 00104'024034-         LDA 1,.MEADR
40 00105'031400          LDA 2,0,3
41 00106'062006A         DOR 0,MCAT
42 00107'065006          DOA 1,MCAT
43 00110'073106          DOCS 2,MCAT
44 00111'063606          SKPDN MCAT
45 00112'000777          JMP .-1
46 00113'054022-         STA 3,MCAKP
47 00114'126520          SUBZL 1,1     ; AC1=1 INDICATES HEADER WAS JUST SENT
48 00115'030033-         LDA 2,.RSTRT
49 00116'006042-         JSR @.TRNSV
50 00117'034022-         LDA 3,MCAKP
51 00120'175400          INC 3,3
52 00121'010007-         ISZ CTR
53 00122'000761          JMP RSRT1
54                  ;    SEND START PULSE TO PREPROCESSOR
55                  ;    RECEIVE 10K VALUES (5K PAIRS) FROM PREPROCESSOR
56                  ;C      LDA 0,MMAX    ; -10,000
57                    ;C      LDA 1,.PIM
58                  ;C      DOC 0,PPRO
59                    ;C      DOR 1,PPRO
```

```
  0005  DPIM
01                ;C      NIOC PPPD    ; START PULSE
02                  ;C      SKPDN PPPD
03                    ;C      JMP .-1
04                ;D      THE FOLLOWING CODE IS TEMPORARILY FOR USE
05                ;D      OF PTR (PAPER TAPE READER) TO INPUT DATA
06                ;D      IGNORE FEED HOLES AT BEGINNING
07                ;D      LOOK FOR 2 FRAMES WITH ALL BITS SET -
08                ;D        THIS INDICATES BEGINNING OF DATA
09                ;D      THEN READ DATA UNTIL 10 CONTINUOUS FEED
10                ;D      HOLES ARE ENCOUNTERED
11                ;
12                ;    READ IN A TAPE OF X,Y COORDINATE POINTS
13 00123'006041-        JSP @,PIMIM
14 00124'040012-        STA 0,PIMCT     ; AC0 CONTAINS NEGATIVE COUNT
15 00125'000401         JMP .+1         ; OF RIM DATA
16 00126'010037-        ISZ FRMCT
17 00127'000401         JMP .+1    ; LEAVE IN AS PROTECTION
18 00130'010040-        ISZ SCNCT
19 00131'000401         JMP .+1    ; LEAVE IN AS PROTECTION
20                ;    SEND THE RIM DATA TO DIM AND WAIT UNTIL DONE
21 00132'030023-        LDA 2,DMMCA
22 00133'006043-        JSR @,SENDR
23                ;    MAIN LOOP PROCESSING
24                ;    READ KEYS
25 00134'060477 MLOOP:READS 0
26 00135'101200         MOVR 0,0
27 00136'101200         MOVR 0,0
28 00137'101202         MOVR 0,0,SZC
29 00140'000774         JMP .-4    ; IF KEY 13 IS SET, WAIT - CHANGE TAPE
30 00141'060477         READS 0    ; IF KEY 15 IS SET, REMOVE THE
31 00142'101202         MOVR 0,0,SZC   ; USER CLOCK & WAIT FOR CONSOLE
32 00143'000432         JMP EXIT    ; COMMAND TO RETURN TO MOOS
33 00144'101203         MOVR 0,0,SNC   ; IS KEY 14 SET?
34 00145'000403         JMP NEWDT    ; NO
35 00146'020012-        LDA 0,PIMCT
36 00147'000403         JMP SKPDT    ; NEW DATA
37                ;    SEND START TO PREPROCESSOR
38                ;    RECEIVE 10K VALUES (5K PAIRS) FROM PREPROCESSOR
39                  ;C      LDA 0,MMAX    ; -10,000
40                ;C      LDA 1,.PIM    ; RIM ADDRESS
41                ;C    DOC 0,PPPD
42                ;C    DOR 1,PPPD
43                ;C      NIOC PPPD    ; START PULE
44                  ;C SKPDN PPPD
45                ;C      JMP .-1
46                ;D    THE FOLLOWING CODE IS TEMPORARY FOR USING PTR
47 00150'006041-NEWDT:JSP @,PIMIN
48 00151'040012-        STA 0,PIMCT    ; NEGATIVE WORD COUNT OF RIM DATA
49 00152'000401 SKPDT:JMP .+1
50 00153'010037-        ISZ FRMCT
51 00154'000401         JMP .+1    ; LEAVE IN AS PROTECTION
52 00155'010040-        ISZ SCNCT
53 00156'000401         JMP .+1    ; LEAVE IN AS PROTECTION
54                ;    SORT THE DATA & SEND TO 4 CAMS
55                ;    CALLING SEQUENCE OF RCSRT
56                ;    AC1= POSITIVE # OF POINTS TO BE SORTED
57                ;    AC2= BASE ADDRESS OF DATA
58 00157'104400         NEG 0,1
59 00160'030011-        LDA 2,.RIM
```

```
  0006  DRIM
01 00161'006044-        JSR @.PCSPT
02 00162'000401         JMP .+1
03                ;     SEND THE PIM DATA TO RECM1
04 00163'030031-        LDA 2,R1MCA
05 00164'006043-        JSR @.SENDP
06                ;     CHECK KEYS
07                ;     IF KEY 0 IS SET, WAIT
08                ;     IF KEY 1 IS SET, DO NOT SEND THIS FRAME
09                ;          TO DIM OR RECM1.  SEND RESETS TO DIM & 4 CAMS & TPKIN, AND
10                ;            START PROCESSING FROM SCRATCH
11 00165'060477         READS 0
12 00166'101102         MOVL 0,0,SZC
13 00167'000776         JMP .-2
14 00170'101102         MOVL 0,0,SZC
15 00171'000643         JMP RESET
16                ;     SEND THE PIM DATA TO THE DIM
17 00172'030023-        LDA 2,DMMCA
18 00173'006043-        JSR @.SENDP
19 00174'000740         JMP MLOOP
20 00175'126520 EXIT:   SUBZL 1,1
21 00176'034002-        LDA 3,SAVE
22 00177'177777         .UCEX
23 00200'054020-RIMIN:  STA 3,SAVER   ; SUBROUTINE TO READ IN UR TO 20,000 FRAMES
24 00201'034011-        LDA 3,.PIM    ; FROM RIR - RETURNS NEGATIVE COUNTER IN AC0
25 00202'126400         SUB 1,1     ; IF UPLOW=0, STORE WORD IN BITS 0-7
26 00203'044017-        STA 1,UPLOW   ; IF UPLOW=1, STORE WORD IN BITS 8-15
27 00204'044014-        STA 1,NONUL   ; IF NONUL=0, ALL NULLS ARE TAKEN TO
28 00205'020005-        LDA 0,M5    ; BE LEADING FEED HOLES
29 00206'040007-        STA 0,CTR   ; IF NONUL=1, WHEN 10 NULLS IN A ROW
30 00207'020016-        LDA 0,MMAX1   ;  ARE DETECTED, READ STOPS
31 00210'040015-        STA 0,CTR1
32 00211'060112         NIOS PTR    ; START PAPER TAPE READER
33 00212'063412 REDIN:  SKPDN PTR
34 00213'000777         JMP .-1
35 00214'060512         DIAS 0,PTR    ; READ IN WORD FROM PTR
36 00215'101004         MOV 0,0,SZR   ; IS IT A NULL WORD?
37 00216'000435         JMP STORE   ; NO
38 00217'125005         MOV 1,1,SNR   ; YES - HAS NON-NULL WORD BEEN DETECTED?
39 00220'000772         JMP REDIN     ; NO - CONTINUE READING
40 00221'030017-        LDA 2,URLOW   ; SHOULD WORD BE STORED IN BITS 0-7
41 00222'151004         MOV 2,2,SZR   ; OR BITS 8-15
42 00223'000407         JMP WHOL1
43 00224'151400         INC 2,2       ; BITS 0-7
44 00225'050017-        STA 2,URLOW
45 00226'101300         MOVS 0,0
46 00227'041400         STA 0,0,3
47 00230'000401         JMP .+1
48 00231'000761         JMP REDIN
49 00232'152400 WHOL1:  SUB 2,2
50 00233'050017-        STA 2,UPLOW   ; BITS 8-15
51 00234'031400         LDA 2,0,3
52 00235'113000         ADD 0,2
53 00236'051400         STA 2,0,3
54 00237'000401         JMP .+1
55 00240'175400         INC 3,3
56 00241'010015-        ISZ CTR1
57 00242'000402         JMP .+2
58 00243'000406         JMP EXCD1
59 00244'151005         MOV 2,2,SNR
```

```
  0007  DPIM
01 00245'010007-        ISZ CTR
02 00246'000744         JMP REDIN
03 00247'060212         NIOC PTR
04 00250'000437         JMP CONT1
05 00251'060212 EXCD1:  NIOC PTR
06 00252'000435         JMP CONT1
07 00253'125004 STORE:  MOV 1,1,SZR     ; FIRST TIME NON-NULL WORD
08 00254'000403         JMP .+3         ; IS DETECTED
09 00255'125400         INC 1,1
10 00256'040010-        STA 1,NONUL
11 00257'030005-        LDA 2,MS    ; RESET NULL COUNTER
12 00260'050007-        STA 2,CTR
13 00261'030017-        LDA 2,UPLOW     ; SHOULD WORD BE STORED IN BITS 0-7
14 00262'151004         MOV 2,2,SZR     ; OR BITS 8-15
15 00263'000407         JMP @HOL2
16 00264'151400         INC 2,2
17 00265'050017-        STA 2,UPLOW     ; BITS 0-7
18 00266'101300         MOVS 0,0
19 00267'041400         STA 0,0,3
20 00270'000401         JMP .+1
21 00271'000721         JMP REDIN
22 00272'152400 @HOL2:  SUB 2,2
23 00273'050017-        STA 2,UPLOW     ; BITS 8-15
24 00274'031400         LDA 2,0,3
25 00275'113000         ADD 0,2
26 00276'100005         COM 2,0,SNR
27 00277'000713         JMP REDIN   ; THIS CODE ALLOWS SKIPPING 177777 START WORD
28 00300'051400         STA 2,0,3
29 00301'000401         JMP .+1
30 00302'175400         INC 3,3
31 00303'010015-        ISZ CTR1
32 00304'000706         JMP REDIN
33 00305'060212 EXCD2:  NIOC PTP
34 00306'000401         JMP CONT1
35 00307'020015-CONT1:  LDA 0,CTR1
36 00310'100400         NEG 0,0
37 00311'024010-        LDA 1,MMAX
38 00312'123000     ADD 1,0
39 00313'002020-        JMP @SAVEP
40 00314'054020-SENDP:  STA 3,SAVEP
41 00315'050046-        STA 2,SVMCA
42                  ;   CREATE THE HEADER BLOCK
43 00316'034034-        LDA 3,.HEADR
44 00317'020032-        LDA 0,PMMCA
45 00320'041401         STA 0,1,3
46 00321'020012-        LDA 0,PIMCT
47 00322'041402         STA 0,2,3
48 00323'020037-        LDA 0,FPMCT
49 00324'041405         STA 0,5,3
50 00325'020040-        LDA 0,SCNCT
51 00326'041406         STA 0,6,3
52                  ;   SEND THE HEADER BLOCK
53 00327'020013-TR1:    LDA 0,MIA
54 00330'024034-        LDA 1,.HEADR
55 00331'030046-        LDA 2,SVMCA
56 00332'062006         DOB 0,MCAT
57 00333'065006         DOA 1,MCAT
58 00334'073106         DOCS 2,MCAT
59 00335'063606         SKPDN MCAT
```

```
  0008  DPIM
01 00336'000777          JMP .-1
02 00337'126520          SUBZL 1,1    ; AC1=1 INDICATES HEADER JUST SENT
03 00340'030035-         LDA 2,.TR1
04 00341'006042-         JSR @.TRNSV
05                     ;  SEND THE DATA BLOCK
06 00342'034065-TR2:     LDA 3,M2
07 00343'020012-         LDA 0,PIMCT
08 00344'163000          ADD 3,0   ; COUNT IS 2 MORE TO ALLOW FOR 0 TRANSFER
09 00345'024011-         LDA 1,.RIM
10 00346'167000          ADD 3,1   ; ADDRESS IS 2 LESS TO ALLOW FOR 0 TRANSFER
11 00347'030046-         LDA 2,SVMCA
12 00350'062004          DOP 0,MCAT
13 00351'045004          DOA 1,MCAT
14 00352'073106          DOCS 2,MCAT
15 00353'063604          SKPDN MCAT
16 00354'000777          JMP .-1
17 00355'126400          SUB 1,1      ; AC1=0 INDICATES DATA JUST SENT
18 00356'030034-         LDA 2,.TR2
19 00357'006042-         JSR @.TRNSV
20 00360'002020-         JMP @SAVER
21                ;   SORT ROUTINE WHICH SORTS RIM DATA & SENDS IT TO THE 4 CAMS
22                ;   AC1 CONTAINS POSITIVE # OF POINTS TO BE SORTED
23                ;  AC2 CONTAINS BASE ADDRESS OF DATA
24 00361'050052-PCSRT:STA 3,SVRTN
25                ;      ZERO OUT TEST SECTION OF BOUNDS BLOCK
26 00362'020072-         LDA 0,M2A
27 00363'040063-         STA 0,CTRI
28 00364'030056-         LDA 3,BNDBS
29 00365'020073-         LDA 0,PA
30 00366'117000          ADD 0,3
31 00367'102400          SUB 0,0
32 00370'041400          STA 0,0,3
33 00371'175400          INC 3,3
34 00372'010063-         ISZ CTRI
35 00373'000775          JMP .-3
36 00374'020058-         LDA 0,CAMAD
37 00375'040055-         STA 0,CAMCT
38 00376'020062-         LDA 0,M4
39 00377'040063-         STA 0,CTRI
40 00400'102400          SUB 0,0
41 00401'040064-         STA 0,FINL
42 00402'176000          ADC 3,3    ; GENERATE A -1
43 00403'167000          ADD 3,1
44 00404'147000          ADD 2,1    ; AC1 CONTAINS FINAL Y LOCATION OF
45 00405'044134-         STA 1,K    ; THIS DATA SET, OR K
46 00406'050053-         STA 2,OTLAD   ; AC2 CONTAINS BASE ADDRESS OF DATA, OR J0
47 00407'030056-         LDA 3,BNDBS
48 00410'050057-         STA 3,BNDPP   ; SET I=0
49 00411'020060-NEXTI:LDA 0,TSS
50 00412'113000          ADD 0,2    ; PREPARE JF
51 00413'132533 ENDTS:SUBZL# 1,2,SNC  ; END OF DATA?
52 00414'002146-         JMP @.NMORE
53                     ; YES
54 00415'021000 GETY: LDA 0,0,2
55                     ; GET Y SUB JF
56 00416'030070-         LDA 3,YMASK
57 00417'163000          AND 3,0
58                     ; STRIP 4 HIGH BITS FROM Y
59 00420'030057-         LDA @3,BNDPR  ; GET BOUND SUB I
```

```
  000R  DPIM
00021'162533        SUBZL# 3,0,SNC  ; IS THE Y VALUE WITHIN THE BOUNDS?
00022'000003        JMP SHORT   ; NO - BACK UP
00023'030061-FULL: LDA 3,T56    ; ADD 256
00024'173000        ADD 3,2     ; TO JF
00025'000766        JMP ENDTS   ; AND LOOP BACK
00026'102120 SHORT:ADCZL 0,0    ; GENERATE -2
00027'113000        ADD 0,2   ; BACK UP 1 Y VALUE
00030'021000        LDA 0,0,2   ; GET Y
00031'054071-       STA 3,SAVE3
00032'030070-       LDA 3,YMASK   ; STRIP OFF 4 HIGH BITS OF Y
00033'163000        AND 3,0
00034'030071-       LDA 3,SAVE3
00035'162533        SUBZL# 3,0,SNC  ; WITHIN BOUNDS?
00036'000770        JMP SHORT   ; NO
00037'030057-GOTIT:LDA 3,BNORB
00040'051020        STA 2,CAMH,3
00041'050067-       STA 2,FINLY   ; FINAL Y ADDRESS FOR THIS CAM
00042'020053-       LDA 1,PTLAD
00043'045010        STA 1,CAML,3
00044'133000        SUB 1,2
00045'151000        INC 2,2
00046'051030        STA 2,NOCAM,3  ; AC2 CONTAINS POSITIVE WORD COUNT
                ; CREATE THE HEADER BLOCK
00047'030030-       LDA 3,.HEADR
00050'020032-       LDA 0,PMMCA   ; SENDING MCA ADDRESS
00051'041001        STA 0,1,3
00052'150000        SEC 2,2
00053'020044-       LDA 0,P5000   ; CHECK # OF WORDS
00054'143000        ADD 2,0
00055'101102        MOVL 0,0,SZC    ; IF > 5000, LIMIT TO 5000
00056'030046-       LDA 2,M5000
00057'051002        STA 2,2,3
00060'050060-       STA 2,WDCNT   ; NEGATIVE WORD COUNT OF BLOCK
                                  ; TO BE SENT TO THE CAM
00061'020037-       LDA 0,FRMCT
00062'041005        STA 0,5,3
00063'020040-       LDA 0,SCNCT
00064'041006        STA 0,6,3
00065'022055-       LDA 0,WCAMCT
00066'041007        STA 0,7,3
              ;  SEND THE HEADER BLOCK
00067'020013-TRC1:LDA 0,MIA
00070'024030-       LDA 1,.HEADR
00071'032055-       LDA 2,@CAMCT
00072'062004        DOB 0,MCAT
00073'065005        DOA 1,MCAT
00074'073104        DOCS 2,MCAT
              ;  SET UP RECEIVE FOR CAM ACKNOWLEDGE
00075'060207 RCC1: NIOC MCAR
00076'020005-       LDA 0,MS
00077'024137-       LDA 1,.READY
00100'062007        DOB 0,MCAR
00101'065107        DOAS 1,MCAR
00102'063A06        SKPDN MCAT
00103'000777        JMP .-1
00104'126520        SUBZL 1,1   ; AC1=1 INDICATES THAT HEADER WAS JUST SENT
00105'030134-       LDA 2,.TRC1
00106'004042-       JSR @,TRNSV
00107'063607        SKPDN MCAR
```

```
  0010  OPTM
01 00510'000777          JMP .-1
02 00511'006051-         JSR @.RCVSV
03 00512'030137-         LDA 2,.READY
04 00513'021001          LDA 0,1,2
05 00514'026055-         LDA 1,#CAMCT
06 00515'104404          SUB 0,1,SZR
07 00516'000400          JMP .    ; READY ACKNOWLEDGE FROM WRONG MACHINE
08 00517'021000          LDA 0,0,2
09 00520'024047-         LDA 1,RDCDE
10 00521'104404          SUB 0,1,SZR
11 00522'000400          JMP .    ; CORRECT MACHINE - WRONG TYPE OF CODE
12 00523'021003          LDA 0,3,2  ; IS HEADER ACCEPTABLE TO CAM?
13 00524'101004          MOV 0,0,SZR
14 00525'000742          JMP TRC1  ; NO - REPEAT TRANSMISSION
15 00526'030065-TRC2:    LDA 3,M2    ; YES - SEND THE DATA BLOCK
16 00527'024053-         LDA 1,DTLAD
17 00530'167000          ADD 3,1
18 00531'030050-         LDA 3,.DTCDE
19 00532'136415          SUB# 1,3,SNR
20 00533'000416          JMP CONT2
21 00534'131000          MOV 1,2
22 00535'050140-         STA 2,SAVPM
23 00536'021000          LDA 0,0,2
24 00537'040141-         STA 0,SAVWD
25 00540'021001          LDA 0,1,2
26 00541'040142-         STA 0,SAVWD+1
27 00542'102520          SUBZL 0,0
28 00543'040143-         STA 0,FIXUP
29 00544'021400          LDA 0,0,3
30 00545'041000          STA 0,0,2
31 00546'021401          LDA 0,1,3
32 00547'041001          STA 0,1,2
33 00550'000403          JMP CONT3
34 00551'102400 CONT2:   SUB 0,0
35 00552'040143-         STA 0,FIXUP
36 00553'000401 CONT3:   JMP .+1
37 00554'030065-         LDA 3,M2
38 00555'020066-         LDA 0,#DCNT
39 00556'163000          ADD 3,0    ; #DCNT IS 2 MORE TO ALLOW
40                                  ; FOR ZERO TRANSFER
41 00557'032055-         LDA 2,#CAMCT
42 00560'062006          DOB 0,MCAT
43 00561'065006          DOA 1,MCAT
44 00562'073106          DOCS 2,MCAT
45 00563'063606          SKPDN MCAT
46 00564'000777          JMP .-1
47 00565'126400          SUB 1,1    ; AC1=0 INDICATES THAT DATA WAS JUST SENT
48 00566'030135-         LDA 2,.TRC2
49 00567'006042-         JSR @.TRNSV
50 00570'020143-         LDA 0,FIXUP
51 00571'101005          MOV 0,0,SNR
52 00572'000406          JMP CONT4
53 00573'030140-         LDA 2,SAVPM
54 00574'020141-         LDA 0,SAVWD
55 00575'041000          STA 0,0,2
56 00576'020142-         LDA 0,SAVWD+1
57 00577'041001          STA 0,1,2
58 00600'010055-CONT4:   ISZ CAMCT
59 00601'010057-         ISZ BNDRR
```

```
  0011  OPIM
01 00602'010063-        ISZ CTRI
02 00603'000402         JMP .+2
03 00604'002052-        JMP @SVRTN    ; EXIT
04 00605'020064-        LDA 0,FINL    ; HAS DATA BEEN EXHAUSTED?
05 00606'101005         MOV 0,0,SNR
06 00607'000405         JMP MORE      ; NO
07 00610'030067-        LDA 2,FINLY   ; YES
08 00611'125400         INC 2,1
09 00612'040053-        STA 1,DTLAD
10 00613'000424         JMP GOTIT
11 00614'024136-MORE:   LDA 1,K
12 00615'030067-        LDA 2,FINLY
13 00616'151400         INC 2,2
14 00617'050053-        STA 2,DTLAD
15 00620'002147-        JMP @.NXTI
16 00621'131000 NMORE:  MOV 1,2       ; IF=K
17 00622'021000         LDA 0,0,2     ; GET Y SUB JF
18 00623'030070-        LDA 3,YMASK
19 00624'163000         AND 3,0
20 00625'036057-        LDA 3,@NDPR   ; GET ROUNDS SUB I
21 00626'162533         SUBZL# 3,0,SNC
22 00627'002150-        JMP @.SHRT
23 00630'010064-        ISZ FINL
24 00631'002153-        JMP @.GTIT
25       000005 READY:  .BLK 5
26       000020 HEADR:  .BLK 20
27       000000'        .END OPIM
```

```
0012  OPIM

ANDRS   000056-     2/13    8/28    8/47
ANDRR   000057-     2/14    8/48    8/59    9/15    10/59   11/20
BOUND   000074-     2/13    2/30
C3MCA   000024-     1/45    2/11
C4MCA   000025-     1/46
C5MCA   000026-     1/47
C6MCA   000027-     1/48
CAMAD   000054-     2/11    8/36
CAMCT   000055-     2/12    8/37    9/39    9/44    10/05   10/41   10/58
CAMM    000020      2/28    9/16
CAML    000010      2/27    9/19
CONT1   000307'     7/04    7/06    7/34    7/35
CONT2   000551'     10/20   10/34
CONT3   000553'     10/33   10/36
CONT4   000600'     10/52   10/58
CTR     000007-     1/32    4/34    4/52    6/29    7/01    7/12
CTR1    000015-     1/38    6/31    6/56    7/31    7/35
CTRT    000063-     2/18    8/27    8/34    8/39    11/01
DMMCA   000023-     1/42    1/44    5/21    6/17
OPIM    000000'     3/20    11/27
OTCDE   000152-     3/21    3/45
OTLAD   000053-     2/10    8/46    9/18    10/16   11/09   11/14
ENDTS   000413'     8/51    9/05
ERR     000012'     3/28    3/32    3/34
EXCD1   000251'     6/58    7/05
EXCD2   000305'     7/33
EXIT    000175'     5/32    6/20
FINL    000064-     2/19    8/41    11/04   11/23
FINLY   000067-     2/22    9/17    11/07   11/12
FIXUP   000143-     3/11    10/28   10/35   10/50
FRMCT   000037-     1/56    4/10    5/16    5/50    7/48    9/35
FULL    000423'     9/03
GETY    000415'     8/54
GOTIT   000437'     3/22    9/15    11/10
HDCDE   000045-     2/03    4/13
HEADR   000637'     1/53    11/26
K       000136-     3/05    8/45    11/11
LOC0    000003-     1/28
LOC1    000004-     1/29
M16     000013-     1/36    4/38    7/53    9/42
M2      000065-     2/20    8/06    10/15   10/37
M20     000072-     2/25    8/26
M4      000062-     2/17    8/38
M5      000005-     1/30    6/28    7/11    9/50
M5000   000145-     3/16    9/31
M6      000006-     1/31    4/33
MCAAD   000021-     1/42    4/35
MCAKP   000022-     1/43    4/37    4/46    4/50
MLOOP   000134'     5/25    6/19
MMAX    000010-     1/33    7/37
MMAX1   000016-     1/39    6/30
MORE    000614'     11/06   11/11
NEWDT   000150'     5/34    5/47
NEXTI   000411'     3/18    8/49
NMORE   000621'     3/17    11/16
NOCAM   000030      2/29    9/22
NONUL   000014-     1/37    6/27    7/10
P5000   000144-     3/15    9/28
PR      000073-     2/26    8/29
```

```
0013  DRIM

PCSRT   000361'     2/02    8/24
RIMCT   000012-     1/35    5/14    5/35    5/48    7/46    8/07
RIMIN   000200'     1/58    6/23
RMMCA   000032-     1/51    7/44    8/25
RRRQ    00002A      1/24
RIMCA   000031-     1/50    6/04
RCC1    000475'     9/4R
RCVSV   000051-X    2/07
RDCDE   000047-     2/05    10/09
READY   000632'     3/06    11/25
REDIN   000212'     6/33    6/39    6/4R    7/02    7/21    7/27    7/32
RESET   000034'     3/56    6/15
RSRT1   000103'     4/36    4/3R    4/53
RSTRT   000102'     1/52    4/37
S10     000001-     1/26    3/24
SAVE    000002-     1/27    3/34    3/37    3/3R    6/21
SAVE3   000071-     2/24    9/0R    R/12
SAVER   000020-     1/41    6/23    7/39    7/40    8/20
SAVRM   000140-     3/07    10/22   10/53
SAVWD   000141-     3/09    10/24   10/2A   10/5A   10/5A
SCNCT   000040-     1/57    4/11    5/18    5/52    7/50    8/37
SENDR   000314'     2/01    7/40
SHORT   000426'     3/1R    9/02    R/06    9/14
SKRDT   000152'     5/36    5/49
STORE   000253'     6/37    7/07
SVMCA   00004A-     2/04    7/41    7/55    8/11
SVRTN   000052-     2/09    8/24    11/03
TSS     000060-     2/15    8/49
TSA     000061-     2/16    9/03
TIMCA   000030-     1/49
TP1     000327'     1/54    7/53
TR2     000342'     1/55    8/0A
TRC1    000467'     3/03    9/42    10/14
TRC2    00052A'     3/04    10/15
TRNSV   000042-X    1/59
UCLOC   00001A'     1/25    3/3R
UPLOW   000017-     1/40    6/2A    6/40    6/44    6/50    7/13    7/17    7/23
WDCNT   000086-     2/21    9/33    10/3R
WHOL1   000232'     6/42    6/49
WHOL2   000272'     7/15    7/22
YMASK   000070-     2/23    8/5A    R/10    11/1R
.DTCD   000050-     2/06    3/47    10/1R
.GTIT   000153-     3/22    11/24
.MEAD   000034-     1/53    4/12    4/39    7/43    7/54    8/24    9/43
.NMAX   000151-     3/20    3/42
.NMOR   000146-     3/17    8/52
.NXTI   000147-     3/18    11/15
.PCSR   000044-     2/02    6/01
.PIM    000011-     1/34    3/52    5/59    6/24    8/09
.RIMI   000041-     1/58    5/13    5/47
.RCVS   000051-     2/07    10/02
.READ   000137-     3/06    8/51    10/03
.RSTR   000033-     1/52    4/4R
.SEND   000043-     2/01    5/22    6/05    6/1R
.SHRT   000150-     3/1R    11/22
.TP1    000035-     1/54    8/03
.TR2    00003A-     1/55    8/19
.TRC1   000134-     3/03    8/57
.TRC2   000135-     3/04    10/48


0014  DRIM

.TRNS   000042-     1/59    4/49    8/04    8/1R    9/5R    10/4R
.UCEX   000177'X    6/22
.UCLO   000000-     1/25    3/25
```

```
  0001  TRNSV
01                      .TITL TRNSV
02                      .ENT TRNSV
03                      .NREL
04                  ;  MCA TRANSMIT SERVICE ROUTINE - NO INTERRUPTS USED
05                  ;    THIS VERSION IS TO BE USED IN THE PIM & DIM, AND IT WILL
06                  ;    TAKE CARE OF THE CASE WHERE THEY MAY SEND THEIR 16 WORD HEADERS
07                  ;    TO THE CAMS BUT THE CAMS ARE SET UP ONLY TO RECEIVE A 5 WORD
08                  ;    ACKNOWLEDGE BLOCK FROM TRACK INITIATION.  IN THAT CASE,
09                  ;    TRANSMITTER COUNT NOT DONE IS IGNORED, & CONTROL RETURNS TO
10                  ;    THE CALLING PROGRAM TO WAIT FOR NEGATIVE ACKNOWLEDGE.
11 00000'054434 TRNSV:STA 3,NRTRN   ; NORMAL RETURN ADDRESS
12 00001'050434       STA 2,SRTRN   ; SPECIAL RETURN ADDRESS - FOR REREAT
13 00002'044441       STA 1,MDIND   ; IF MDIND=0, DATA OR ACKNOWLEDGE BLOCK WAS
14                                  ;     JUST SENT
15                                  ; IF MDIND=1, HEADER BLOCK WAS JUST SENT
16 00003'060405       DIA 0,MCAT
17 00004'040435       STA 0,TWDAD   ; PRESENT CONTENTS OF ADDRESS COUNTER
18 00005'061406       DIB 0,MCAT
19 00006'040434       STA 0,TWDCT   ; PRESENT CONTENTS OF WORD COUNTER
20                                  ; THIS SHOULD BE ZERO
21 00007'062405       DIC 0,MCAT
22 00010'040424       STA 0,STATT
23 00011'024426       LDA 1,TIMDT
24 00012'107405       AND 0,1,SNR
25 00013'000404       JMP NEXT
26 00014'060206       NIOC MCAT
27 00015'000401       JMP .+1
28 00016'001000       JMP 0,2   ; RETURN TO RETRANSMIT
29 00017'024421 NEXT: LDA 1,CNTDT
30 00020'107404       AND 0,1,SZR
31 00021'000410       JMP CLEAR
32 00022'024421       LDA 1,MDIND
33 00023'125005       MOV 1,1,SNR
34 00024'063077       HALT          ; TRANSMITTER COUNT NOT DONE ON DATA OR ACKNOWLEDGE
35 00025'024415       LDA 1,TWDCT   ; TAKE PRESENT CONTENTS OF WORD COUNTER
36 00026'030416       LDA 2,P11     ; ADD TO +11.
37 00027'133004       ADD 1,2,SZR   ; IF RESULT=0, THEN THIS WAS CASE WHERE 5 WORDS
38                                  ;  OF A 16 WORD HEADER WERE ACCEPTED
39 00030'063077       HALT          ; IF RESULT IS NOT ZERO, THEN THIS IS LEGITIMATE
40                                  ;  TRANSMIT COUNT NOT DONE.
41 00031'060204 CLEAR:NIOC MCAT
42 00032'000401       JMP .+1
43 00033'001400       JMP 0,3   ; NORMAL RETURN
44 00034'000000 NRTRN:0
45 00035'000000 SRTRN:0
46 00036'000000 STATT:0
47 00037'000010 TIMDT: 10B12
48 00040'000002 CNTDT: 10B14
49 00041'000000 TWDAD:0
50 00042'000000 TWDCT:0
51 00043'000000 MDIND:0
52 00044'000013 P11:11.
53                      .END  ; END OF TRNSV
```

```
  0002  TRNSV

CLEAR   000031'     1/31    1/41
CNTDT   000040'     1/29    1/48
MDIND   000043'     1/13    1/32    1/51
NEXT    000017'     1/25    1/29
NRTRN   000034'     1/11    1/44
P11     000044'     1/36    1/52
SRTRN   000035'     1/12    1/45
STATT   000036'     1/22    1/46
TIMDT   000037'     1/23    1/47
TRNSV   000000'     1/11
TWDAD   000041'     1/17    1/49
TWDCT   000042'     1/19    1/35    1/50
```

```
                                TRNSV1
  0001  TRNSV
01                      .TITL TRNSV
02                      .ENT TRNSV
03                      .NREL
04                  ;  MCA TRANSMIT SERVICE ROUTINE - NO INTERRUPTS USED
05 00000'050424 TRNSV:STA 3,NRTRN  ; NORMAL RETURN ADDRESS
06 00001'050424       STA 2,SRTRN  ; SPECIAL RETURN ADDRESS - FOR REPEAT
07 00002'060006       DIA 0,MCAT
08 00003'040425       STA 0,TWDAD  ; PRESENT CONTENTS OF ADDRESS COUNTER
09 00004'061406       DIB 0,MCAT
10 00005'040425       STA 0,TWDCT  ; PRESENT CONTENTS OF WORD COUNTER
11                                 ; THIS SHOULD BE ZERO
12 00006'062006       DIC 0,MCAT
13 00007'040417       STA 0,STATT
14 00010'024417       LDA 1,TIMDT
15 00011'107405       AND 0,1,SNR
16 00012'000404       JMP NEXT
17 00013'060206       NIOC MCAT
18 00014'000401       JMP .+1
19 00015'001000       JMP 0,2  ; RETURN TO RETRANSMIT
20 00016'024412 NEXT: LDA 1,CNTDT
21 00017'107405       AND 0,1,SNR
22 00020'000400       JMP .  ; XMTR COUNT NOT DONE
23 00021'060206       NIOC MCAT
24 00022'000401       JMP .+1
25 00023'001400       JMP 0,3  ; NORMAL RETURN
26 00024'000000 NRTRN:0
27 00025'000000 SRTRN:0
28 00026'000000 STATT:0
29 00027'000010 TIMDT: 1B12
30 00030'000002 CNTDT: 1B14
31 00031'000000 TWDAD:0
32 00032'000000 TWDCT:0
33                      .END  ; END OF TRNSV
```

```
  0002  TRNSV

CNTDT   000030'     1/29    1/30
NEXT    000016'     1/16    1/20
NRTRN   000024'     1/05    1/26
SRTRN   000025'     1/06    1/27
STATT   000026'     1/13    1/28
TIMDT   000027'     1/14    1/29
TRNSV   000000'     1/05
TWDAD   000031'     1/08    1/31
TWDCT   000032'     1/10    1/32
```

```
  0001  RCVSV                    RCVSV1
01                      .TITL RCVSV
02                      .ENT RCVSV
03                      .NREL
04                  ;  MCA RECEIVE SERVICE ROUTINE - NO INTERRUPTS USED
05 00000'060407 RCVSV:DIA 0,MCAR
06 00001'040417         STA 0,RWDAD   ; PRESENT CONTENTS OF ADDRESS COUNTER
07 00002'061407         DIR 0,MCAR
08 00003'040416         STA 0,RWDCT   ; PRESENT CONTENTS OF WORD COUNTER
09                                    ; SHOULD BE ZERO
10 00004'062407         DIC 0,MCAR
11 00005'040410         STA 0,STATR
12 00006'020410         LDA 1,TIMDT
13 00007'107404         AND 0,1,SZR
14 00010'000400         JMP .         ; RCVR TIME OUT
15 00011'020406         LDA 1,CNTDR
16 00012'107405         AND 0,1,SNR
17 00013'000400         JMP .  ; RCVR COUNT NOT DONE
18 00014'001400         JMP 0,3
19 00015'000000 STATR:0
20 00016'000010 TIMDT: 1612
21 00017'000001 CNTDR: 1B15
22 00020'000000         RWDAD:0
23 00021'000000   RWDCT:0
24                      .END   ; END OF RCVSV
```

```
  0002  RCVSV

CNTDR   000017'     1/15    1/21
RCVSV   000000'     1/05
RWDAD   000020'     1/06    1/22
RWDCT   000021'     1/08    1/23
STATR   000015'     1/11    1/19
TIMDT   000016'     1/12    1/20
```

```
                                DDIM12A
  0001  DDIM
01                      .TITL DDIM
02                       .ENT DDIM
03                       .EXTN TRNSV,PCVSV
04                ;    DEMONSTRATION VERSION OF DIM PROGRAM, WHICH
05                ;    RECEIVES DATA FROM THE PIM, SORTS IT, AND
06                ;    SENDS IT TO THE  4 CAMS.
07                ;    KEYS
08                ;    THE THRESHOLD VALUE WILL BE READ IN FROM THE
09                ;    KEYS ONCE FOR EACH FRAME.
10                ;    THE SORT ROUTINE TESTS THE AMPLITUDE OF EACH
11                ;    Y VALUE & DISCARDS ANY PAIR WHOSE Y AMPLITUDE IS LESS
12                ;    THAN THE THRESHOLD VALUE READ IN FROM THE KEYS.
13                ;    DESIGNED 23 JANUARY 1975
14                         .7REL
15 00000-177760 M1A: -1A.
16 00001-000303'.HEADR: HEADR
17 00002-177777 .PCVSV:PCVSV
18 00003-177777 .TRNSV: TRNSV
19 00004-100004 RDCDE: -32764.
20 00005-000004-.DTCDE: DTCDE
21 00006-100001 DTCDE: -32767.
22 00007-000000       0
23                      ; THE ABOVE TWO WORDS ARE ADDED TO ALL DATA TRANSMISSIONS
24 00010-100003 RDCDE: -32765.
25 00011-000000 .PIMS:0
26 00012-000000 .PIM:0
27 00013-000334'.READY: READY
28 00014-120000 R4MCA: 120000
29 00015-110000 D4MCA: 110000
30 00016-020000 C3MCA: 020000
31 00017-040000 C4MCA: 040000
32 00020-030000 C5MCA: 030000
33 00021-060000 C6MCA: 060000
34 00022-000016-CAMAD: C3MCA
35 00023-000000 CAMCT:0
36 00024-177774 M4: -4
37 00025-000000 CTRI:0
38 00026-000000 FINL:0
39 00027-000360 YLMSK:360
40 00030-007777 YHMSK:7777
41 00031-000000 WDCNT:0
42 00032-000000 CTRP:0
43 00033-177776 M2:-2
44 00034-000000 CTR:0
45 00035-177750 M24: -24.
46 00036-000051-ANDRS:ROUND
47 00037-000010 PR: 8.
48 00040-000000 ANDPR:0
49 00041-000000 DTHS:0
50 00042-177777 M1:-1
51 00043-000000 DTCTR:0
52 00044-000000 NXTAD:0
53 00045-000231'.TRC1:TRC1
54 00046-000270'.TRC2:TRC2
55 00047-177773 M5:  -5
56 00050-000010'.RSET:RESET
57        000010 CAML=10
58        000020 CAMH=20
59        000030 NOCAM=30
```

```
  0002  ODIM
01 00051-000400 BOUND:256.
02 00052-001000        512.
03 00053-001400        768.
04 00054-002000 1024.
05 00055-000000        0
06 00056-000000        0
07 00057-000000        0
08 00060-000000        0
09 00061-000000        0   ; STORAGE FOR CAM BASE ADDRESS - CAML
10 00062-000000        0
11 00063-000000        0
12 00064-000000        0
13 00065-000000        0
14 00066-000000        0
15 00067-000000        0
16 00070-000000        0
17 00071-000000        0   ; STORAGE FOR CAM FINAL ADDRESS - CAMH
18 00072-000000        0
19 00073-000000        0
20 00074-000000        0
21 00075-000000        0
22 00076-000000        0
23 00077-000000       0
24 00100-000000        0
25 00101-000000        0   ; STORAGE FOR CAM WORD COUNT - NOCAM
26 00102-000000        0
27 00103-000000        0
28 00104-000000        0
29 00105-000000        0
30 00106-000000        0
31 00107-000000       0
32 00110-000000        0
33 00111-011610 P5000:5000.
34 00112-000404 .NMAX:404
35 00113-000000 AMP:0
36                     .NREL
37 00000'000401 ODIM: JMP .+1
38 00001'062877        DICC 0,CPU
39                ;    FIND OUT NMAX ADDRESS AND SET .PIMS AND .PIM ADDRESSES
40 00002'034112-       LDA 3,.NMAX
41 00003'031400        LDA 2,0,3
42 00004'050011-       STA 2,.PIMS
43 00005'151400        INC 2,2
44 00006'151400        INC 2,2
45 00007'050012-       STA 2,.PIM
46 00010'063206 RESET:NIOC MCAT
47 00011'063506        SKPBZ MCAT
48 00012'000400        JMP .
49 00013'063706        SKPDZ MCAT
50 00014'000400        JMP .
51                ;  UNLOCK RECEIVER
52 00015'063207 UNLCK:NIOC MCAR
53 00016'063507        SKPBZ MCAR
54 00017'000400        JMP .
55 00020'063707        SKPDZ MCAR
56 00021'000400        JMP .
57                ; SET UP MCA RECEIVE FOR HEADER
58 00022'020000-       LDA 0,M1B
59 00023'024001-       LDA 1,.HEADR
```

```
  0003  DDIM
01 00024'062007         DOB 0,MCAR
02 00025'065107         DOAS 1,MCAR
03 00026'063607         SKPDN MCAR
04 00027'000777         JMP .-1
05 00030'004002-        JSR @.RCVSV
06 00031'030001-        LDA 2,.HEADR
07 00032'021000         LDA 0,0,2
08 00033'024004-        LDA 1,MDCDE
09 00034'106404         SUB 0,1,SZR
10                ;  IS IT A HEADER?
11 00035'000760         JMP UNLCK   ; NO
12 00036'021001 TSRST:LDA 0,1,2   ; YES - IS IT A RESET HEADER?
13 00037'101415         INC# 0,0,SNR
14 00040'000750         JMP RESET
15 00041'024014-        LDA 1,DMMCA   ; IS THIS A PIM HEADER?
16 00042'106404         SUB 0,1,SZR
17 00043'000752         JMP UNLCK   ; NO
18 00044'021002         LDA 0,2,2   ; YES - TAKE WORD COUNT
19 00045'040031-        STA 0,WDCNT
20 00046'024034-        LDA 1,M2    ; SUBTRACT 2 TO ALLOW FOR
21 00047'123000         ADD 1,0     ; 2 EXTRA WORDS
22 00050'024011-        LDA 1,.PIMS   ; WORD ADDRESS IS SET AT
23 00051'062007         DOB 0,MCAR
24 00052'065107         DOAS 1,MCAR   ; 2 BEFORE PIM BLOCK
25 00053'063607         SKPDN MCAR
26 00054'000777         JMP .-1
27 00055'004002-        ISR @.RCVSV
28 00056'030011-        LDA 2,.PIMS
29 00057'021000         LDA 0,0,2
30 00060'024004-        LDA 1,MDCDE   ; IS THIS A HEADER?
31 00061'106414         SUB# 0,1,SZR
32 00062'000414         JMP TSDAT     ; NO
33 00063'020000-        LDA 0,M16     ; YES - TRANSFER FIRST 16
34 00064'040034-        STA 0,CTR     ; WORDS TO HEADER BLOCK
35 00065'030011-        LDA 2,.PIMS ; & TEST IF IT IS RESET
36 00066'034001-        LDA 3,.HEADR
37 00067'021000 LOOP1:LDA 0,0,2
38 00070'041400         STA 0,0,3
39 00071'151400         INC 2,2
40 00072'175400         INC 3,3
41 00073'010034-        ISZ CTR
42 00074'000773         JMP LOOP1
43 00075'000741         JMP TSRST
44 00076'024006-TSDAI:LDA 1,DTCDE
45 00077'106414         SUB# 0,1,SZR
46 00100'000400         JMP .   ; NOT DATA CODE
47                ;  SORT ROUTINE SORTS THE PIM DATA BY
48                ;   1) CHECKING EACH Y'S AMPLITUDE AGAINST THE
49                ;      AMPLITUDE READ IN VIA THE KEYS
50                ;   2) DISCARDING ANY X,Y PAIR WHOSE AMPLITUDE
51                ;      IS LESS THAN THAT IN THE KEYS
52                ;   3) SEARCHING THE DATA LIST REMAINING & SENDING
53                ;      ALL PAIRS WITH
54                ;        Y < 256 TO CAM3
55                ;        256 <= Y < 512 TO CAM4
56                ;        512 <= Y < 768 TO CAM5
57                ;        768 <= Y < 1024 TO CAM6
58 00101'020022-        LDA 0,CAMAD
59 00102'040023-        STA 0,CAMCT
```

```
  0004  00IM
01 00103'020024-        LDA 0,M4
02 00104'040025-        STA 0,CTRI
03 00105'102400         SUB 0,0
04 00106'040026-        STA 0,FINL
05                ;   ZERO OUT TEST SECTION OF ROUND BLOCK
06 00107'020035-        LDA 0,M24
07 00110'040034-        STA 0,CTR
08 00111'034036-        LDA 3,SNDRS
09 00112'020037-        LDA 0,PA
10 00113'117000         ADD 0,3
11 00114'102400         SUB 0,0
12 00115'041400         STA 0,0,3
13 00116'175400         INC 3,3
14 00117'010034-        ISZ CTR
15 00120'000775         JMP .-3
16 00121'024031-        LDA 1,#DCNT
17 00122'044032-        STA 1,CTRP  ; POINT COUNTER
18 00123'034036-        LDA 3,SNDRS
19 00124'054040-        STA 3,ANDRR
20 00125'030012-        LDA 2,.PIM  ; AC2 CONTAINS BASE
21 00126'050041-        STA 2,OTBS  ; BASE FOR OUTPUT ARRAY - REAL
22 00127'020042-        LDA 0,M1
23 00130'143000         ADD 2,0
24 00131'040020         STA 0,20,0  ; BASE FOR OUTPUT ARRAY - LOC. 20
25 00132'102400         SUB 0,0
26 00133'040043-        STA 0,OTCTR  ; OUTPUT ARRAY COUNTER
27 00134'060477         READS 0  ; READ KEYS TO GET AMPLITUDE
28 00135'040113-        STA 0,AMP
29 00136'021001 SLOOP:LDA 0,1,2  ; TAKE Y
30 00137'101300         MOVS 0,0
31 00140'024027-        LDA 1,YLMSK
32 00141'123400         AND 1,0
33 00142'024113-        LDA 1,AMP   ; LOAD AC1 WITH AMPLITUDE
34 00143'122523         SUBZL 1,0,SNC  ; IS YAMP-KEYS<0?
35 00144'000412         JMP GETY        ; NO - CONTINUE
36 00145'151400         INC 2,2         ; YES
37 00146'151400         INC 2,2
38 00147'010032-        ISZ CTRP
39 00150'000402         JMP .+2
40 00151'000400         JMP .   ; ODD # OF WORDS
41 00152'010032-        ISZ CTRP
42 00153'000743         JMP SLOOP
43 00154'010026-        ISZ FINL
44 00155'000424         JMP SENDC
45 00156'021001 GETY: LDA 0,1,2  ; GET Y
46 00157'024030-        LDA 1,YHMSK
47 00160'107400         AND 0,1
48 00161'036040-        LDA #3,ANDRR  ; GET ROUND SUB I
49 00162'166533         SUBZL# 3,1,SNC  ; IS Y<ROUND?
50 00163'000416         JMP SENDC       ; NO - SEND TO CAM
51 00164'025000         LDA 1,0,2       ; YES - GET X
52 00165'046020         STA 1,@20,0  ; PUT X IN OUTPUT ARRAY
53 00166'042020         STA 0,@20,0  ; PUT Y IN OUTPUT ARRAY
54 00167'010043-        ISZ OTCTR
55 00170'010043-        ISZ OTCTR
56 00171'151400         INC 2,2
57 00172'151400         INC 2,2
58 00173'010032-        ISZ CTRP
59 00174'000402         JMP .+2
```

```
  0005  ODIM
01 00175'000400        JMP .  ; ODD # OF WORDS?
02 00176'010032-       ISZ CTRP
03 00177'000737        JMP SLOOP
04 00200'010026-       ISZ FINL
05 00201'050044-SENDC:STA 2,NXTAD
06 00202'034040-       LDA 3,ANDPR
07 00203'024042-       LDA 1,M1
08 00204'133000        ADD 1,2
09 00205'051020        STA 2,CAMH,3  ; FINAL Y ADDRESS FOR THIS CAM
10 00206'024041-       LDA 1,DTAS
11 00207'045010        STA 1,CAML,3  ; BASE X ADDRESS FOR THIS CAM
12 00210'030043-       LDA 2,DTCTR
13 00211'051030        STA 2,NOCAM,3  ; # OF DATA WORDS
14                ;  CREATE THE HEADER BLOCK
15 00212'034001-       LDA 3,.HEADR
16 00213'020015-       LDA 0,DMMCA
17 00214'041401        STA 0,1,3
18 00215'150400        NEG 2,2
19 00216'020111-       LDA 0,PS000
20 00217'143000        ADD 2,0
21 00220'101103        MOVL 0,0,SNC
22 00221'000404        JMP CONT1
23 00222'030111-       LDA 2,PS000
24 00223'050043-       STA 2,DTCTR
25 00224'150400        NEG 2,2
26 00225'000401 CONT1:JMP .+1
27 00226'051402        STA 2,2,3
28                ;   LEAVE FRAME # AND SCAN # AS IS
29 00227'022023-       LDA 0,@CAMCT
30 00230'041407        STA 0,7,3
31                ;  SEND THE HEADER BLOCK
32 00231'020000-TRC1: LDA 0,M16
33 00232'024001-       LDA 1,.HEADR
34 00233'032023-       LDA 2,@CAMCT
35 00234'062006        DOB 0,MCAT
36 00235'065005        DOA 1,MCAT
37 00236'073106        DOCS 2,MCAT
38                ;  SET UP RECEIVE FROM CAM ACKNOWLEDGE
39 00237'060207 RCC1: NIOC MCAR
40 00240'020047-       LDA 0,M5
41 00241'024013-       LDA 1,.READY
42 00242'062007        DOB 0,MCAR
43 00243'065107        DOAS 1,MCAR
44 00244'063606        SKPDN MCAT
45 00245'000777        JMP .-1
46 00246'126520        SUBZL 1,1    ; AC1=1 INDICATES THAT HEADER WAS JUST SENT
47 00247'030045-       LDA 2,.TRC1
48 00250'006003-       JSR @.TRNSV
49 00251'063607        SKPDN MCAR
50 00252'000777        JMP .-1
51 00253'006002-       JSP @.RCVSV
52 00254'030013-       LDA 2,.READY
53 00255'021001        LDA 0,1,2
54 00256'026023-       LDA 1,@CAMCT
55 00257'106404        SUB 0,1,SZR
56 00260'000400        JMP .        ; READY ACKNOWLEDGE FROM WRONG MACHINE
57 00261'021000        LDA 0,0,2
58 00262'024010-       LDA 1,RDCDE
59 00263'106404        SUB 0,1,SZR
```

```
  0006  DDIM
01 00264'000400          JMP .  ; CORRECT MACHINE - WRONG TYPE OF CODE
02 00265'021003          LDA 0,3,2  ; IS HEADER ACCEPTABLE TO CAM?
03 00266'101004          MOV 0,0,SZR
04 00267'000742          JMP TRC1  ; NO - REREAT TRANSMISSION
05 00270'034033-TRC2: LDA 3,M2     ; YES - SEND THE DATA BLOCK
06 00271'024041-         LDA 1,DTRS
07 00272'167000          ADD 3,1
08 00273'034005-         LDA 3,.DTCDF
09 00274'131000          MOV 1,2
10 00275'021400          LDA 0,0,3
11 00276'041000          STA 0,0,2
12 00277'021401          LDA 0,1,3
13 00300'041001          STA 0,1,2
14 00301'034033-         LDA 3,M2
15 00302'020043-         LDA 0,DTCTR
16 00303'100400          NEG 0,0
17 00304'143000          ADD 3,0
18 00305'032023-         LDA 2,@CAMCT
19 00306'062006          DOB 0,MCAT
20 00307'065006          DOA 1,MCAT
21 00310'073106          DOCS 2,MCAT
22 00311'063606          SKPDN MCAT
23 00312'000777          JMP .-1
24 00313'126400          SUB 1,1      ; AC1=0 INDICATES THAT DATA WAS JUST SENT
25 00314'030046-         LDA 2,.TRC2
26 00315'006003-         JSR @.TRNSV
27 00316'010023-         ISZ CAMCT
28 00317'010040-         ISZ ANDRR
29 00320'030044-         LDA 2,NXTAD
30 00321'050041-         STA 2,DTRS
31 00322'020042-         LDA 0,M1
32 00323'143000          ADD 2,0
33 00324'040020          STA 0,20,0
34 00325'102400          SUB 0,0
35 00326'040043-         STA 0,DTCTR
36 00327'010025-         ISZ CTRI
37 00330'000402          JMP .+2
38 00331'002050-         JMP @.RSET
39 00332'020026-         LDA 0,FINL
40 00333'101005          MOV 0,0,SNR
41 00334'000602          JMP SLOOP
42 00335'000644          JMR SENDC
43       000005 READY: .BLK 5
44       000020 HEADR: .BLK 20
45       000000'       .END DDIM
```

```
0007  DDIM

AMP       000113-     2/35    4/28    4/33
ANDBS     00003A-     1/46    4/0A    4/1A
ANDPR     000040-     1/48    4/19    4/4A    5/06    6/28
ADUND     000051-     1/46    2/01
C3MCA     00001A-     1/30    1/34
C4MCA     000017-     1/31
C5MCA     000020-     1/32
CAMCA     000021-     1/33
CAMAD     000022-     1/34    3/5A
CAMCT     000023-     1/35    3/59    5/29    5/3A    5/5A    6/1A    6/27
CAMM      000020      1/58    5/09
CAML      000010      1/57    5/11
CONT1     000225'     5/22    5/26
CTP       00003A-     1/44    3/34    3/41    4/07    4/1A
CTPI      000025-     1/37    4/02    6/36
CTPP      000032-     1/42    4/17    4/3A    4/41    4/5A    5/02
DDIM      000000'     2/37    6/45
DMMCA     000015-     1/29    5/16
DTCDE     00000A-     1/20    1/21    3/44
FINL      00002A-     1/38    4/0A    4/43    5/0A    6/39
GETY      000156'     4/35    4/45
HDCDE     00000A-     1/19    3/0A    3/30
HEADR     0003A3'     1/14    6/44
LOOP1     000067'     3/37    3/42
M1        000042-     1/50    4/22    5/07    6/31
M16       000000-     1/15    2/5A    3/33    5/32
M2        000033-     1/43    3/20    6/05    6/1A
M2A       000035-     1/45    4/0A
M4        00002A-     1/36    4/01
M5        000047-     1/55    5/40
MDCAM     000030      1/59    5/13
NXTAD     000044-     1/52    5/05    6/29
OTBS      000041-     1/49    4/21    5/10    6/06    6/30
OTCTR     0000A3-     1/51    4/2A    4/5A    4/55    5/12    5/2A    6/15    6/35
P5000     000111-     2/33    5/19    5/23
PA        000037-     1/47    4/09
PMMCA     00001A-     1/24    3/15
PCC1      000237'     5/39
PCVSV     000002-X    1/17
PDCDE     000010-     1/24    5/5A
READY     000336'     1/27    6/43
RESET     000010'     1/56    2/4A    3/1A
SENDC     000201'     4/44    4/5A    5/05    6/42
SLOOP     000136'     4/29    4/42    5/03    6/41
TPC1      000231'     1/53    5/32    6/04
TPC2      000270'     1/54    6/05
TRNSV     000003-X    1/18
TSDAT     00007A'     3/32    3/44
TSRST     000036'     3/12    3/43
UNLCK     000015'     2/52    3/11    3/17
WDCNT     000031-     1/41    3/19    4/16
YHMSK     000030-     1/40    4/46
YLMSK     000027-     1/39    4/31
.DTCD     000005-     1/20    6/0A
.HEAD     000001-     1/16    2/59    3/06    3/36    5/15    5/33
.NMAX     000112-     2/34    2/40
.PIM      000012-     1/26    2/45    4/20
.PIMS     000011-     1/25    2/42    3/22    3/28    3/35
.WCVS     000002-     1/17    3/05    3/27    5/51


0008  DDIM

.WEAD     000013-     1/27    5/41    5/52
.WSET     000050-     1/56    6/3A
.TPC1     000045-     1/53    5/47
.TPC2     00004A-     1/54    4/25
.TRNS     000003-     1/18    5/4A    6/24
```

```
0001  DCAM3                    DCAM3RA
01                      .TITL DCAM3
02                       .ENT DCAM3
03                       .EXTN CTLKR,LK1,TB1AD,TB2AD,CTTR1,CTTR2
04                       .EXTN CNCLS,RCVSV,TRNSV
05                       .EXTN LKDVF
06                  ;   DEMONSTRATION VERSION OF CAM PROGRAM, CALLING
07                  ;   STAR CANCELLATION ROUTINE, SENDING LEAKERS TO
08                  ;   RECM1, AND USING AN INTERRUPT SERVICE ROUTINE
09                  ;   TO HANDLE MCA TRANSFERS
10                  ;   DESIGNED 14 JANUARY 1975
11                       .TREL
12 00000-030000 CMCA:030000   ; THIS CAM MCA ADDRESS WILL VARY
13 00001-177760 M16:-16.
14 00002-000542'.DIMHD:.DIMHD
15 00003-000000 ENTRY: 0
16 00004-000000 HDRD: 0
17 00005-100004 RDCDE:-32764.
18 00006-110000 DMCA:110000
19 00007-000510'.REDY:REDY
20 00010-000074'.TR5:TNGD   ; TRANSMIT NO GOOD REPLY TO SENDING MACHINE
21 00011-177773 M5:-5
22 00012-011610 M5000:5000.
23 00013-166170 M5000:-5000.
24 00014-177777 .CNCLS:CNCLS
25 00015-177777 .LK1:LK1
26 00016-177777 .TB1AD:TB1AD
27 00017-177777 .TB2AD:TB2AD
28 00020-177777 .CTR1:CTTR1
29 00021-177777 .CTR2:CTTR2
30 00022-177777 .CTLKR:CTLKR
31 00023-177777 .LKDVF:LKDVF
32 00024-120000 RMCA:120000
33 00025-050000 XMCA:050000
34 00026-000135'.TR1:TRDYD ; TRANSMIT READY TO DIM
35 00027-000254'.TR2:TRDYR ; TRANSMIT READY TO PIM
36 00030-000340'.TR3:TRECM ; TRANSMIT RECM HEADER TO RECM
37 00031-000352'.TR4:TRECD ; TRANSMIT LEAKER DATA TO RECM
38 00032-000213'.TR6:TNGR   ; TRANSMIT NO GOOD REPLY TO SENDING MACHINE
39 00033-000000 .DIM4:0
40 00034-000542'.PIMHD:.DIMHD
41 00035-000000 .PIM4:0
42 00036-000522'.RECHD:.RECHD
43 00037-100001 DICDE:-32767.
44 00040-000044'.LSN1:LISN1
45 00041-000000 .DIMS:0
46 00042-000000 .PIMS:0
47 00043-177776 M2:-2
48 00044-000004-.HDRD:HDRD
49 00045-000000 SVMCA:0
50 00046-177777 .RCVSV:RCVSV
51 00047-177777 .TRNSV:TRNSV
52 00050-170000 MCAMK:170000
53 00051-000404 .NMAX:404
54 00052-100000 TMCA:100000
55 00053-000402'.TR7:TTIH     ; TRANSMIT TRACK INITIATION HEADER
56 00054-000470'.TR8:TTID     ; TRANSMIT TRACK INITIATION LEAKERS
57 00055-000515'.RDY1:REDY1
58 00056-100003 RDCDE:-32765.
59 00057-000441'.TR9:TNGTI    ; TRANSMIT NO GOOD REPLY TO SENDING MACHINE
```

```
  0002  DCAM3
01                                ; IT SHOULD HAVE RECEIVED AN ACKNOWLEDGE FROM T.I.
02 00000-000133 P91:91.
03                        .NREL
04 00000'062677 DCAM3:DICC 0,CRU
05 00001'030051-        LDA 3,.NMAX     ; FIND OUT NMAX ADDRESS & SET
06 00002'031000         LDA 2,0,3         ; .DIMS, .DIMAD, .PIMS, & .PIMAD ADDRESSES
07 00003'050041-        STA 2,.DIMS
08 00004'151000         INC 2,2
09 00005'151000         INC 2,2
10 00006'050033-        STA 2,.DIMAD
11 00007'020012-        LDA 0,R5000
12 00010'113000         ADD 0,2
13 00011'050042-        STA 2,.PIMS
14 00012'151000         INC 2,2
15 00013'151000         INC 2,2
16 00014'050035-        STA 2,.PIMAD
17 00015'000401         JMP .+1
18 00016'030007-        LDA 3,.RFDY
19 00017'060207         NIOC MCAR
20 00020'062407         DIC 0,MCAR      ; FIND OUT MCA CODE FOR
21 00021'020051-        LDA 1,MCAMK     ; THIS MACHINE
22 00022'107400         AND 0,1
23 00023'040000-        STA 1,CMCA      ; STORE MCA CODE IN CURRENT
24 00024'045401         STA 1,1,3       ; MCA VARIABLE & READY BLOCK
25 00025'000401         JMP .+1
26 00026'000401 RESET:JMP .+1     ; COME HERE WHEN RESET HEADER
27                                ; IS RECEIVED, INDICATING THAT PROCESSING
28                                ; IS TO BE RESTARTED.
29 00027'102400         SUB 0,0
30 00030'040003-        STA 0,ENTRY     ; WHEN ENTRY = 0, DIM HEADER IS EXPECTED
31                                      ; WHEN ENTRY = 1, PIM HEADER IS EXPECTED
32 00031'040004-        STA 0,HORD      ; WHEN HORD = 0, HEADER IS EXPECTED
33                                      ; WHEN HORD = 1, DATA IS EXPECTED
34 00032'060206         NIOC MCAT
35 00033'063506         SKRBZ MCAT
36 00034'000400         JMP .
37 00035'063706         SKRDZ MCAT
38 00036'000400         JMP .
39 00037'060207         NIOC MCAR
40 00040'063507         SKPBZ MCAR
41 00041'000400         JMP .
42 00042'063707         SKPDZ MCAR
43 00043'000400         JMP .
44 00044'060207 LISN1:NIOC MCAR
45 00045'020001-        LDA 0,M16   ; SET UP RECEIVE FOR HEADER FROM DIM
46 00046'020002-        LDA 1,.DIMHD    ; ADDRESS OF DIM HEADER BLOCK
47 00047'062007         DOB 0,MCAR
48 00050'065107         DOAS 1,MCAR
49 00051'063607         SKPDN MCAR
50 00052'000777         JMP .-1
51 00053'006000-        JSR @.RCVSV
52 00054'000401         JMP .+1
53 00055'030002-        LDA 2,.DIMHD
54 00056'025000         LDA 1,0,2       ; TAKE FIRST WORD FROM HEADER BLOCK
55 00057'030005-        LDA 3,HDCDE     ; IS IT -32764.?
56 00060'136400         SUB 1,3,SZR
57 00061'000763         JMP LISN1     ; IF WORD 1 IS NOT HEADER CODE,
58                                    ; GO BACK TO LISTEN
59 00062'025001         LDA 1,1,2     ; TAKE SECOND WORD FROM HEADER BLOCK
```

```
  0003  DC4M3
01 00063'125415          INC# 1,1,SNR    ; IS IT -1?
02 00064'000742          JMP RESET       ; IF SO, RESET
03 00065'034006-         LDA 3,DMCA
04 00066'136405          SUB 1,3,SNR     ; IS THIS HEADER FROM DIM?
05 00067'000420          JMP QI1   ; YES - CONTINUE
06 00070'044045-         STA 1,SVMCA  ; SAVE MCA CODE
07 00071'034007-         LDA 3,.REDY  ; NO - SEND BACK REPLY OF
08 00072'102520          SUBZL 0,0  ;  NOT ACCEPTABLE
09 00073'041403          STA 0,3,3
10 00074'020011-TNGO:    LDA 0,M5 ; TRANSMIT ACKNOWLEDGE TO
11 00075'024007-         LDA 1,.REDY ; MACHINE WHICH SENT HEADER
12 00076'030045-         LDA 2,SVMCA
13 00077'062006          DOR 0,MCAT
14 00100'065004          DOA 1,MCAT
15 00101'073104          DOCS 2,MCAT
16 00102'063606          SKPDN MCAT
17 00103'000777          JMP .-1
18 00104'030010-         LDA 2,.TR5
19 00105'006047-         JSR @.TRNSV
20 00106'000736          JMP LISN1       ; NO, GO BACK TO LISTEN
21 00107'000401 QI1:     JMP .+1
22 00110'025007           LDA 1,7,2    ; TAKE QUADRANT INDICATOR
23 00111'020000-         LDA 0,CMCA
24 00112'104404          SUB 0,1,SZR
25 00113'000400          JMP .    ; HEADER BLOCK HAS WRONG QUADRANT INDICATOR
26 00114'010004-         ISZ WORD
27                  ;   SET UP RECEIVE FOR DATA FROM DIM
28 00115'021002          LDA 0,2,2    ; TAKE WORD COUNT FROM HEADER BLOCK
29 00116'024012-         LDA 1,P5000  ; TEST INCOMING WORD COUNT
30 00117'107000          ADD 0,1
31 00120'125103          MOVL 1,1,SZC
32 00121'020013-         LDA 0,M5000
33 00122'042021-         STA 0,2,CTR2  ; PUT IN CANCEL'S TABLE 2 WORD COUNT
34 00123'030043-         LDA 2,M2
35 00124'103000          ADD 2,0
36 00125'024033-         LDA 1,.DIMAD  ; TAKE ADDRESS OF DIM DATA
37 00126'046017-          STA 1,2,TR2AD ; PUT IN CANCEL'S TABLE 2 DATA ADDRESS
38 00127'024041-         LDA 1,.DIMS  ; STARTING DATA ADDRESS MINUS 2
39 00130'062007          DOR 0,MCAR
40 00131'065107          DOAS 1,MCAR
41 00132'034007-         LDA 3,.REDY
42 00133'102400          SUB 0,0
43 00134'041403          STA 0,3,3  ; HEADER IS ACCEPTABLE
44 00135'020011-TRDYD:   LDA 0,M5   ; TRANSMIT READY MESSAGE TO DIM
45 00136'024007-         LDA 1,.REDY
46 00137'030004-         LDA 2,DMCA
47 00140'062006          DOR 0,MCAT
48 00141'065004          DOA 1,MCAT
49 00142'073104          DOCS 2,MCAT
50 00143'063606          SKPDN MCAT
51 00144'000777          JMP .-1
52 00145'030024-         LDA 2,.TR1
53 00146'006047-         JSR @.TRNSV
54 00147'063607          SKPDN MCAR
55 00150'000777          JMP .-1
56 00151'006046-         JSR @.RCVSV
57                  ;   CHECK THAT THIS IS A DATA BLOCK
58 00152'030041-         LDA 2,.DIMS
59 00153'025000          LDA 1,0,2
```

```
  0000  DCAM3
01 00150'030037-        LDA 3,DTCDE
02 00151'136400         SUB 1,3,SZR
03 00152'000000         JMP .   ; WRONG BLOCK FOLLOWED DIM HEADER
04 00153'010003-        ISZ ENTRY
05 00154'102400         SUB 0,0
06 00155'040004-        STA 0,WORD
07 00156'000001         JMP .+1
08 00157'060207 LISN2:NIOC MCAR
09 00160'020001-        LDA 0,M16  ; SET UP RECEIVE FOR HEADER FROM PIM
10 00161'024030-        LDA 1,.PIMHD
11 00162'062007         DOB 0,MCAR
12 00163'065107         DOAS 1,MCAR
13 00164'063607         SKPDN MCAR
14 00165'000777         JMP .-1
15 00166'006004-        JSR @.RCVSV
16 00167'000001         JMP .+1
17 00170'030030-        LDA 2,.PIMHD
18 00171'025000         LDA 1,0,2   ; TAKE FIRST WORD FROM HEADER BLOCK
19 00172'034005-        LDA 3,HDCDE   ;IS IT -32764.?
20 00173'136404         SUB 1,3,SZR
21 00174'000763         JMP LISN2   ; IF IT IS NOT  HEADER CODE, LISTEN AGAIN
22 00175'025001         LDA 1,1,2   ; TAKE SECOND WORD FROM HEADER BLOCK
23 00176'125415         INC# 1,1,SNR   ; IS IT -1?
24 00177'000623         JMP RESET
25 00200'030024-        LDA 3,PMCA
26 00201'136405         SUB 1,3,SNR   ; IS THIS HEADER FROM PIM?
27 00202'000020         JMP QI2  ; YES - CONTINUE
28 00203'040045-        STA 1,SVMCA  ; SAVE MCA CODE
29 00204'030007-        LDA 3,.REDY  ; NO - SEND BACK REPLY OF
30 00205'102520         SUBZL 0,0    ; NOT ACCEPTABLE
31 00206'041003         STA 0,3,3
32 00207'020011-TNGP: LDA 0,M5  ; TRANSMIT ACKNOWLEDGE TO
33 00210'024007-        LDA 1,.REDY  ; MACHINE WHICH SENT HEADER
34 00211'030045-        LDA 2,SVMCA
35 00212'062006         DOB 0,MCAT
36 00213'065006         DOA 1,MCAT
37 00214'073106         DOCS 2,MCAT
38 00215'063606         SKPDN MCAT
39 00216'000777         JMP .-1
40 00217'030032-        LDA 2,.TR6
41 00220'006007-        JSR @.TRNSV
42 00221'000736         JMP LISN2   ; NO, GO BACK TO LISTEN
43 00222'000001 QI2:  JMP .+1
44 00223'025007         LDA 1,7,2   ; TAKE QUADRANT INDICATOR
45 00224'020000-        LDA 0,CMCA
46 00225'106400         SUB 0,1,SZR
47 00226'000000         JMP .   ;  QUADRANT INDICATOR IS WRONG
48 00227'010004-        ISZ WORD
49                ;   SET UP RECEIVE FOR DATA FROM PIM
50 00230'021002         LDA 0,2,2   ; TAKE WORD COUNT FROM HEADER BLOCK
51 00231'024012-        LDA 1,M5000  ; TEST INCOMING WORD COUNT
52 00232'107000         ADD 0,1
53 00233'125102         MOVL 1,1,SZC
54 00234'020013-        LDA 0,M5000
55 00235'042020-        STA 0,@.CTR1   ; PUT IN CANCEL'S TABLE 1 WORD COUNT
56 00236'030003-        LDA 2,M2
57 00237'143000         ADD 2,0
58 00240'024035-        LDA 1,.PIMAD ; TAKE PIM DATA ADDRESS
59 00241'044016-        STA 1,@.TP1AD ; PUT IN CANCEL'S TABLE 1 DATA ADDRESS
```

```
  0005  DCAM3
01 00246'020042-       LDA 1,.PIMS
02 00247'062007        DOB 0,MCAP
03 00250'065107        DOAS 1,MCAR
04 00251'034007-       LDA 3,.REDY
05 00252'102400        SUB 0,0
06 00253'041403        STA 0,3,3  ; HEADER IS ACCEPTABLE
07 00254'020011-TRDYP:LDA 0,MS   ; TRANSMIT READY TO PIM
08 00255'024007-       LDA 1,.REDY
09 00256'030024-       LDA 2,PMCA
10 00257'062006        DOB 0,MCAT
11 00260'065006        DOA 1,MCAT
12 00261'073106        DOCS 2,MCAT
13 00262'063606        SKPDN MCAT
14 00263'000777        JMP .-1
15 00264'030027-       LDA 2,.TR2
16 00265'006047-       JSR @.TRNSV
17 00266'063607        SKPDN MCAR
18 00267'000777        JMP .-1
19 00270'006046-       JSR @.RCVSV
20                ;   CHECK THAT THIS IS A DATA BLOCK
21 00271'030042-       LDA 2,.PIMS
22 00272'025000        LDA 1,0,2
23 00273'034037-       LDA 3,DTCDE
24 00274'136400        SUB 1,3,SZR
25 00275'000400        JMP .    ; WRONG BLOCK FOLLOWED PIM HEADER
26 00276'102400        SUB 0,0
27 00277'040003-       STA 0,ENTRY
28 00300'040004-       STA 0,WORD
29 00301'006010-       JSR @.CNCLS   ; CALL STAR CANCELLATION ROUTINE
30 00302'030036-       LDA 2,.RECHD
31 00303'020005-       LDA 0,MDCDE
32 00304'041000        STA 0,0,2
33 00305'020000-       LDA 0,CMCA
34 00306'041001        STA 0,1,2
35 00307'022022-       LDA 0,@.CTLKR
36                ;   DOUBLE CTLKR VARIABLE
37 00310'101124        MOVZL 0,0,SZR
38 00311'100400        NEG 0,0
39 00312'041002        STA 0,2,2
40 00313'034034-       LDA 3,.PIMHD
41 00314'021402        LDA 0,2,3   ; -WORD COUNT OF PIM DATA
42 00315'041004        STA 0,4,2
43 00316'021405        LDA 0,5,3   ; FRAME # OF PIM DATA
44 00317'041005        STA 0,5,2
45 00320'021406        LDA 0,6,3
46 00321'041006        STA 0,6,2
47 00322'034002-       LDA 3,.DIMHD
48 00323'021402        LDA 0,2,3   ; - WORD COUNT OF DIM DATA
49 00324'041010        STA 0,10,2
50 00325'021405        LDA 0,5,3
51 00326'041011        STA 0,11,2
52 00327'021406        LDA 0,6,3
53 00330'041012        STA 0,12,2
54 00331'021410        LDA 0,10,3   ; DISK ADDRESS POINTER OF DIM
55 00332'041013        STA 0,13,2
56 00333'102400        SUB 0,0
57 00334'041014        STA 0,14,2
58 00335'041015        STA 0,15,2
59 00336'041016        STA 0,16,2
```

```
  0006  DCAM3
01 00337'041017         STA 0,17,2
02                ;  TRANSMIT RECM HEADER TO RECM1
03 00340'020001-TRECM:LDA 0,M16
04 00341'024036-        LDA 1,.RECMO
05 00342'030025-        LDA 2,RMCA
06 00343'062006         DOB 0,MCAT
07 00344'065006         DOA 1,MCAT
08 00345'073106         DOCS 2,MCAT
09 00346'063606         SKPDN MCAT
10 00347'000777         JMP .-1
11 00350'030030-        LDA 2,.TR3
12 00351'006047-        JSR @.TRNSV
13                ;  TRANSMIT LIST OF LEAKERS
14 00352'022022-TRECD:LDA 0,@.CTLKR
15 00353'101120         MOVZL 0,0
16 00354'100400         NEG 0,0
17 00355'030043-        LDA 2,M2
18 00356'143000         ADD 2,0
19 00357'024015-        LDA 1,.LK1
20 00360'147000         ADD 2,1
21 00361'030025-        LDA 2,RMCA
22 00362'062006         DOB 0,MCAT
23 00363'065006         DOA 1,MCAT
24 00364'073106         DOCS 2,MCAT
25 00365'063606         SKPDN MCAT
26 00366'000777         JMP .-1
27 00367'030031-        LDA 2,.TR4
28 00370'006047-        JSR @.TRNSV
29                ;   TRANSMIT RECM HEADER TO TRACK INITIATION MACHINE
30                ;    IF LEAKER OVERFLOW INDICATOR IS SET, SEND ONLY
31                ;    91 X,Y PAIRS TO TRACK INITIATION MACHINE.  THIS
32                ;    WILL ELIMINATE THE LAST 9 X,Y PAIRS WHICH ARE A
33                ;    LINE SIMULATED FOR DISPLAY PURPOSES.
34 00371'022023-        LDA 0,@.LKOVF
35 00372'101005         MOV 0,0,SNR   ; IS LEAKER OVERFLOW COUNTER SET?
36 00373'000407         JMP TTIM      ; NO - PROCEED
37 00374'030036-        LDA 2,.RECMO  ; YES - SET UP TO TRANSMIT ONLY
38 00375'020060-        LDA 0,M91     ;       91 X,Y PAIRS
39 00376'042022-        STA 0,@.CTLKR
40 00377'101120         MOVZL 0,0
41 00400'100400         NEG 0,0
42 00401'041002         STA 0,2,2
43 00402'020001-TTIM:LDA 0,M16
44 00403'024036-        LDA 1,.RECMO
45 00404'030052-        LDA 2,TMCA
46 00405'062006         DOB 0,MCAT
47 00406'065006         DOA 1,MCAT
48 00407'073106         DOCS 2,MCAT
49                ;  SET UP RECEIVE FOR ACKNOWLEDGE FROM TRACK INITIATION
50 00410'060207         NIOC MCAR
51 00411'020011-        LDA 0,M5
52 00412'024055-        LDA 1,.RDY1
53 00413'062007         DOB 0,MCAR
54 00414'065107         DOAS 1,MCAR
55 00415'063606         SKPDN MCAT
56 00416'000777         JMP .-1
57 00417'030053-        LDA 2,.TR7
58 00420'006047-        JSR @.TRNSV
59 00421'063607 CKRCV:SKPDN MCAR
```

```
  0007  DCAM3
01 00422'000777          JMP .-1
02 00423'006000-         JSR @.RCVSV
03 00424'030055-         LDA 2,.RDY1
04 00425'021000          LDA 0,0,2    ; TEST CONTENTS OF RDY1 BLOCK FOR
05 00426'024056-         LDA 1,RDCDE    ; CORRECT CONTENTS
06 00427'106405          SUB 0,1,SNR
07 00430'000431          JMP CKADD     ; CONTENTS IS CORRECT - CHECK MCA ADDRESS
08 00431'024005-         LDA 1,HDCDE   ; CONTENTS IS NOT CORRECT - IS IT HEADER?
09 00432'106404          SUB 0,1,SZR
10 00433'063077          HALT          ; NOT HEADER OR READY BLOCK - ILLEGAL
11 00434'034007-         LDA 3,.RDY    ; YES - IT IS HEADER
12 00435'102520          SUBZL 0,0     ; NOT ACCEPTABLE CODE IN READY BLOCK
13 00436'041003          STA 0,3,3
14 00437'021001          LDA 0,1,2
15 00440'040045-         STA 0,SVMCA   ; SAVE MCA ADDRESS  OF WRONG MACHINE
16 00441'020011-TNGTI:LDA 0,M5
17 00442'024007-         LDA 1,.RDY
18 00443'030045-         LDA 2,SVMCA
19 00444'062006          DOB 0,MCAT    ; TRANSMIT NOT ACCEPTABLE READY BLOCK
20 00445'065006          DOA 1,MCAT    ; TO THAT MACHINE
21 00446'073106          DOCS 2,MCAT
22 00447'060207          NIOC MCAR     ; LISTEN FOR RDY1 ACKNOWLEDGE AGAIN
23 00450'020011-         LDA 0,M5
24 00451'024055-         LDA 1,.RDY1
25 00452'062007          DOB 0,MCAR
26 00453'065107          DOAS 1,MCAR
27 00454'063406          SKPDN MCAT    ; WAIT FOR TRANSMIT TO BE DONE
28 00455'000777          JMP .-1
29 00456'030057-         LDA 2,.IRQ
30 00457'006047-         JSR @.IRNSV
31 00460'000701          JMP CKRCV     ; CHECK RECEIVER DONE & CONTENTS OF RDY1 BLOCK
32 00461'021001 CKADD:LDA 0,1,2
33 00462'024052-         LDA 1,TMCA
34 00463'106404          SUB 0,1,SZR
35 00464'063077          HALT          ; READY REPLY FROM WRONG MACHINE
36 00465'021003          LDA 0,3,2       ; IS HEADER ACCEPTABLE TO TRKIN?
37 00466'101004          MOV 0,0,SZR
38 00467'000713          JMP TIIH
39                ;  SEND LEAKERS TO TRKIN
40 00470'022023-TIID: LDA 0,@.CTLKR
41 00471'101120          MOVZL 0,0
42 00472'100400          NEG 0,0
43 00473'030043-         LDA 2,M2
44 00474'103000          ADD 2,0
45 00475'024015-         LDA 1,.LK1
46 00476'103000          ADD 2,1
47 00477'030052-         LDA 2,TMCA
48 00500'062006          DOB 0,MCAT
49 00501'065006          DOA 1,MCAT
50 00502'073106          DOCS 2,MCAT
51 00503'063406          SKPDN MCAT
52 00504'000777          JMP .-1
53 00505'030054-         LDA 2,.IRR
54 00506'006047-         JSR @.IRNSV
55 00507'002040-         JMP @.LSN1
56 00510'100003 RDY:  -32765.    ; BLOCK TO TRANSMIT TO DIM & PIM
57 00511'030000          030000
58 00512'000000          0
59 00513'000000          0
```

```
  0008  DCAM3
01 00514'000000          0
02          000005 RDY1: .BLK 5  ; BLOCK FOR RECEIVING MESSAGES FROM TRACK INITIATION MACHINE
03          000020 RECHO: .BLK 20
04          000020 DIMHO: .BLK 20
05          000020 PIMHO: .BLK 20
06          000000'      .END DCAM3
```

```
0000  DCAM3

CKADD   000A81'     7/07    7/32
CKRCV   000821'     6/59    7/31
CMCA    000000-     1/12    2/23    3/23    4/45    5/33
CNCLS   000014-X    1/24
CTLKR   000022-X    1/30
CTTR1   000020-X    1/28
CTTR2   000021-X    1/29
DCAM3   000000'     3/04    8/06
DIMHD   000542'     1/14    8/04
DMCA    000006-     1/18    3/03    3/46
DTCDE   000037-     1/43    4/01    5/23
ENTRY   000003-     1/15    2/30    4/04    5/27
HDCDE   000005-     1/17    2/55    4/19    5/31    7/08
HDRD    000004-     1/16    1/48    2/32    3/26    4/06    4/48    5/28
LISN1   000040'     1/44    2/44    2/57    3/20
LISN2   0001A3'     4/08    4/21    4/42
LK1     000015-X    1/25
LKOVF   000023-X    1/31
M1A     000001-     1/13    2/45    4/09    6/03    6/43
M2      000043-     1/47    3/34    4/54    6/17    7/43
M5      000011-     1/21    3/10    3/44    4/32    5/07    6/51    7/16    7/23
M5000   000013-     1/23    3/32    4/54
MCAMK   000050-     1/52    2/21
P5000   000012-     1/22    3/11    3/29    4/51
PR1     000060-     2/02    6/38
RIMHD   0005A2'     1/40    8/05
RMCA    000020-     1/32    4/25    5/09
SI1     000107'     3/05    3/21
SI2     000226'     4/27    4/43
RCVSV   000086-X    1/50
RDCDE   000056-     1/58    7/05
RECHD   000522'     1/42    8/03
REDY    000510'     1/19    7/54
REDY1   000515'     1/57    8/02
RESET   000024'     2/26    3/02    4/24
RMCA    000025-     1/33    6/05    6/21
SVMCA   000045-     1/49    3/04    3/12    4/28    4/34    7/15    7/18
TB1AD   00001A-X    1/26
TB2AD   000017-X    1/27
TMCA    000052-     1/54    6/45    7/33    7/47
TNGD    000074'     1/20    3/10
TNGP    000213'     1/38    4/32
TNGTI   000441'     1/59    7/16
TRDYD   000135'     1/34    3/44
TRDYP   000254'     1/35    5/07
TRECD   000352'     1/37    6/14
TRECH   000340'     1/36    6/03
TRNSV   000047-X    1/51
TTID    000470'     1/56    7/40
TTIM    000402'     1/55    6/36    6/43    7/38
.CNCL   000014-     1/24    5/29
.CTR1   000020-     1/28    4/55
.CTR2   000021-     1/29    3/33
.CTLK   000022-     1/30    5/35    6/14    6/39    7/40
.DIMA   000033-     1/39    2/10    3/36
.DIMH   000002-     1/14    2/46    2/53    5/47
.DIMS   000041-     1/45    2/07    3/38    3/5A
.HDRD   000048-     1/48
.LK1    000015-     1/25    6/19    7/45
```

```
0010  GCAM3

.LKOV   000023-  1/31   6/34
.LSN1   000040-  1/44   7/55
.NMAX   000051-  1/53   2/05
.PIMA   000035-  1/41   2/16   4/58
.PIMH   000034-  1/40   4/10   4/17   5/40
.PIMS   000042-  1/46   2/13   5/01   5/21
.PCVS   000046-  1/50   2/51   3/56   4/15   5/19   7/02
.PDY1   000055-  1/57   6/52   7/03   7/24
.PECH   000036-  1/42   5/30   6/04   6/37   6/44
.SFDY   000007-  1/19   2/1A   3/07   3/11   3/41   3/45   4/29   4/33
                 5/04   5/0A   7/11   7/17
.TR1A   00001A-  1/2A   4/59
.TR2A   000017-  1/27   3/37
.TR1    000026-  1/34   3/52
.TR2    000027-  1/35   5/15
.TR3    000030-  1/36   6/11
.TR4    000031-  1/37   6/27
.TR5    000010-  1/20   3/1A
.TR6    000032-  1/38   4/40
.TR7    000053-  1/55   6/57
.TR8    000058-  1/56   7/53
.TR9    000057-  1/59   7/29
.TRNS   000047-  1/51   3/19   3/53   4/41   5/16   6/12   6/28   6/58
                 7/30   7/54
```

```
                                CNCLSTR34
  0001  CNCLS
01                      .TITL CNCLSTP   ; STAR CANCELLATION ROUTINE
02                         .ENT CNCLS,LK1,TR1AD,TR2AD,CTTR1,CTTR2
03                         .ENT CTLKR
04                         .ENT LKOVF
05                   ;   NEW VERSION DESIGNED 30 DECEMBER 1974
06                   ;   CHANGE TO USE 34 Y ROWS OF 64 X VALUES
07                   ;   ADDED 13 JANUARY 1975
08                         .ZREL
09 00000-000000 CTLKR:0
10 00001-000000 SAVE:0
11        000001 LKOVF: .BLK 1   ; LEAKER OVERFLOW INDICATOR
12 00003-000000 OVLAP:0
13 00004-000041 P33:33.
14                         .NREL
15        000001 TR1AD: .BLK 1    ; ADDRESS OF TABLE 1 DATA
16        000001 TR2AD: .BLK 1    ; ADDRESS OF TABLE 2 DATA
17        000001 CTTR1: .BLK 1    ; COUNT OF TABLE 1 DATA
18        000001 CTTR2: .BLK 1    ; COUNT OF TABLE 2 DATA
19 00004'005445'LK1AD:LK1
20 00005'177700 M64:    -64.
21 00006'000000 CTPC:   0
22 00007'000042 P34:    34.
23 00010'000302'.YOCT:YOCT
24 00011'000305'.PRY1:PRVY1
25 00012'000306'.PRY2:PRVY2
26 00013'000002 .P2:2
27 00014'000003-.OVL1:OVLAP
28 00015'000271'.CTRR:CTTR
29 00016'000273'.CTR1:CTTR1
30 00017'000554'.LKOV:LEKOV
31 00020'177634 M100:-100.
32 00021'054001-CNCLS:STA 3,SAVE
33 00022'126000         ADC 1,1    ; -1
34 00023'020755         LDA 0,TR1AD
35 00024'123000         ADD 1,0
36 00025'040020         STA 0,20,0    ; PLACE TABLE 1 ADDRESS IN AUTO
37                                      ; INCREMENTING LOCATION 20
38 00026'020753         LDA 0,TR2AD
39 00027'123000         ADD 1,0
40 00030'040021         STA 0,21,0    ; PLACE TABLE 2 ADDRESS IN AUTO
41                                      ; INCREMENTING LOCATION 21
42 00031'020753         LDA 0,LK1AD    ; PLACE LEAK LIST ADDRESS IN AUTO
43                                      ; INCREMENTING LOCATION 22
44 00032'123000         ADD 1,0
45 00033'040022         STA 0,22,0
46 00034'126400         SUB 1,1
47 00035'046753         STA 1,@.YOCT    ; 0 TO Y OCTANT
48 00036'046753         STA 1,@.PRY1    ; 0 TO PREVIOUS Y1 VALUE
49 00037'046753         STA 1,@.PRY2    ; 0 TO PREVIOUS Y2 VALUE
50 00040'044000-        STA 1,CTLKR    ; 0 TO LEAKER COUNT
51 00041'044002-        STA 1,LKOVF   ; 0 TO LEAKER OVERFLOW INDICATOR
52 00042'020751         LDA 1,.P2
53 00043'046751         STA 1,@.OVL1    ; 2 TO COUNTER OF OVERLAPPED
54                                      ; X,Y COORDINATES
55                   ;   TEST TABLE 1 & TABLE 2 COUNTERS
56                   ;   IF TABLE 2 COUNT=0, EXIT
57                   ;   IF TABLE 2 COUNT IS NOT EQUAL TO 0
58                   ;      AND TABLE 1 COUNT = 0.  TRANSFER ALL
59                   ;      TABLE 2 VALUES TO LEAK LIST AND EXIT
```

```
  0002  CNCLS
01 00044'020737          LDA 0,CTTR2
02 00045'101005          MOV 0,0,SNR
03 00046'001400          JMP 0,3
04 00047'030751          LDA 2,M100
05 00050'050000-         STA 2,CTLKR
06 00051'024731          LDA 1,CTTR1
07 00052'125005          MOV 1,1,SNR
08 00053'000404          JMP TRNFR
09 00054'046741          STA 1,@.CTRR    ; CTTR1 VALUE
10 00055'042741          STA 0,@.CTR1    ; CTTR2 VALUE
11 00056'000417          JMP CLRL1
12 00057'040727 TRNFR:STA 0,CTRC
13 00060'022021          LDA 0,@21,0
14 00061'042022          STA 0,@22,0
15 00062'010000-         ISZ CTLKR
16 00063'000402          JMP .+2
17 00064'002733          JMP @.LKDN
18 00065'010721          ISZ CTRC
19 00066'000772          JMP .-6
20 00067'020000-         LDA 0,CTLKR
21 00070'024730          LDA 1,M100
22 00071'124400          NEG 1,1
23 00072'123000          ADD 1,0
24 00073'040000-         STA 0,CTLKR
25 00074'001400          JMP 0,3
26                ;  FILL 64 BY 34 ARRAY WITH BIT PATTERNS
27                ;  REPRESENTATIVE OF EACH X,Y PAIR IN TABLE 1
28                ; CLEAR THE 64 BY 34 ARRAY
29 00075'020710 CLRL1:LDA 0,M64
30 00076'040710          STA 0,CTRC
31 00077'034710          LDA 3,P34
32 00100'126400          SUB 1,1
33 00101'030562          LDA 2,ROWAD
34 00102'045000 LOOP:    STA 1,0,2
35 00103'045001          STA 1,1,2
36 00104'045002          STA 1,2,2
37 00105'045003          STA 1,3,2
38 00106'045004          STA 1,4,2
39 00107'045005          STA 1,5,2
40 00110'045006          STA 1,6,2
41 00111'045007          STA 1,7,2
42 00112'045010          STA 1,10,2
43 00113'045011          STA 1,11,2
44 00114'045012          STA 1,12,2
45 00115'045013          STA 1,13,2
46 00116'045014          STA 1,14,2
47 00117'045015          STA 1,15,2
48 00120'045016          STA 1,16,2
49 00121'045017          STA 1,17,2
50 00122'045020          STA 1,20,2
51 00123'045021          STA 1,21,2
52 00124'045022          STA 1,22,2
53 00125'045023          STA 1,23,2
54 00126'045024          STA 1,24,2
55 00127'045025          STA 1,25,2
56 00130'045026          STA 1,26,2
57 00131'045027          STA 1,27,2
58 00132'045030          STA 1,30,2
59 00133'045031          STA 1,31,2
```

```
  0003  CNCLS
01 00134'045032          STA 1,32,2
02 00135'045033          STA 1,33,2
03 00136'045034          STA 1,34,2
04 00137'045035          STA 1,35,2
05 00140'045036          STA 1,36,2
06 00141'045037          STA 1,37,2
07 00142'045040          STA 1,40,2
08 00143'045041          STA 1,41,2
09 00144'173000          ADD 3,2
10 00145'010641          ISZ CTRC
11 00146'000734          JMP LOOP
12 00147'036020 LOOP1:LDA 3,@20,0    ; TABLE 1 X VALUE
13 00150'022020          LDA 0,@20,0    ; TABLE 1 Y VALUE
14 00151'024534           LDA 1,PRVY1
15 00152'106405          SUB 0,1,SNR
16 00153'000416          JMP SAMY1
17 00154'024513          LDA 1,MASK3
18 00155'107520          ANDZL 0,1
19 00156'125100          MOVL 1,1
20 00157'125100          MOVL 1,1
21 00160'125300          MOVS 1,1
22 00161'030521          LDA 2,YOCT
23 00162'132405          SUB 1,2,SNR
24 00163'000402          JMP CONT1
25                    ; OCTANT HAS CHANGED
26                    ;  KEEP TRACK OF HOW MANY VALUES IN NEXT OCTANT HAVE
27                    ;  Y=0 AND Y=1.  FILL BIT PATTERN MATRIX WITH THESE X,Y
28                    ;  PAIRS, SUBTRACT # OF X&Y VALUES ADVANCED INTO NEXT
29                    ;  OCTANT AND GO TO LOOP2 FOR CANCELLATION
30                    ;  THE LOGIC FOR THIS HAS BEEN MOVED TO HIGHER CORE
31                    ;  DUE TO ADDRESSING CONSIDERATIONS.
32 00164'002525          JMP @.SPECL
33 00165'040520 CONT1:STA 0,PRVY1
34 00166'030502          LDA 2,MASK4
35 00167'113400          AND 0,2
36 00170'050507          STA 2,Y1    ; Y1 VALUE
37 00171'024475 SAMY1: LDA 1,MASK2
38 00172'167400          AND 3,1
39 00173'044510          STA 1,BITX1
40 00174'024471          LDA 1,MASK1
41 00175'167520          ANDZL 3,1
42 00176'125100          MOVL 1,1
43 00177'125100          MOVL 1,1
44 00200'125100          MOVL 1,1
45 00201'125300          MOVS 1,1
46 00202'000401          JMP .+1
47                    ; AC1 CONTAINS X1 COORDINATE
48 00203'030461          LDA 3,STRAD    ; AC3 CONTAINS ADDRESS OF STAR TABLE
49 00204'030473          LDA 2,Y1     ; AC2 CONTAINS STAR TABLE OFFSET BY Y1 COORDINATE
50 00205'000401          JMP .+1
51 00206'173000          ADD 3,2
52 00207'021000          LDA 0,0,2
53 00210'123000          ADD 1,0
54 00211'000401          JMP .+1
55 00212'024462          LDA 1,PTIAD
56 00213'030470          LDA 2,BITX1
57 00214'034456          LDA 3,WHERE
58 00215'000401          JMP .+1
59 00216'157000          ADD 2,3
```

```
  0004  CNCLS
01 00217'000401          JMP .+1
02 00220'007400          JSP @0,3
03 00221'000401          JMP .+1
04 00222'010447          ISZ CTR
05 00223'000402          JMP .+2
06 00224'000400          JMP .  ; 000 = OF WORDS
07 00225'010444          ISZ CTP
08 00226'000721          JMP LOOP1
09                ;  START OF CANCELLATION
10 00227'036021 LOOP2:LDA 3,@21,0    ; TABLE 2 X VALUE
11 00230'022021          LDA 0,@21,0    ; TABLE 2 Y VALUE
12 00231'024455          LDA 1,PPVY2
13 00232'106405          SUB 0,1,SNR
14 00233'000510          JMP SAMY2
15 00234'024433          LDA 1,MASK3
16 00235'107520          ANDZL 0,1
17 00236'125100          MOVL 1,1
18 00237'125100          MOVL 1,1
19 00240'125300          MOVS 1,1
20 00241'030441          LDA 2,YOCT
21 00242'132405          SUB 1,2,SNR
22 00243'000465          JMP CONT2
23                ; OCTANT HAS CHANGED IN TABLE 2 - CHECK TO SEE
24                ; IF Y COORDINATE VALUE = 0, AND NEW OCTANT=PRIOR OCTANT
25                ; PLUS 1.  IF THAT IS THE CASE, LET Y2 VALUE=33
26                ; AND DON'T CHANGE YOCT.  THIS WILL ALLOW FOR THE TWO
27                ; LINE OVERLAP.  WHEN Y COORDINATE VALUE BECOMES >0,
28                ; THEN BACK TRACK IN LOCATION 21 THE # OF X,Y COORDINATES
29                ; THAT HAVE BEEN ADVANCED SINCE THE OCTANT CHANGE
30                ;PRVY2 HAS NOT BEEN SAVED WHEN OCTANT CHANGES ON PURPOSE
31 00244'054434          STA 3,X2
32 00245'176520          SUBZL 3,3    ; +1
33 00246'173004          ADD 3,2,SZR
34 00247'000444          JMP CHOCT
35 00250'034420          LDA 3,MASK4
36 00251'117404          AND 0,3,SZR
37 00252'000441          JMP CHOCT
38 00253'024003-         LDA 1,OVLAP
39 00254'125400          INC 1,1
40 00255'125400          INC 1,1
41 00256'044003-         STA 1,OVLAP
42 00257'020004-         LDA 0,P33
43 00260'040421          STA 0,Y2
44 00261'034417          LDA 3,X2
45 00262'000461          JMP SAMY2
46 00263'001123'ROWAD: ROW1
47 00264'0010A1'STRAD: STARS
48 00265'001760 MASK1: 1760
49 00266'000017 MASK2: 17
50 00267'000740 MASK3: 740
51 00270'000037 MASK4: 37
52 00271'000000 CTP:   0
53 00272'005343'WHERE: JMPLS
54 00273'000000 CTP1:  0
55 00274'005323'PT1AD: PATN1
56 00275'005423'PT2AD: PATN2
57 00276'000000 X1:    0
58 00277'000000 Y1:    0
59 00300'000000 X2:    0
```

```
  0005  CNCLS
01 00301'000000 Y2:    0
02 00302'000000 YOCT:  0
03 00303'000000 BITX1: 0
04 00304'000000 BITX2: 0
05 00305'000000 RRVY1: 0
06 00306'000000 RRVY2: 0
07 00307'000002 P2:    2
08 00310'000040 R32:   32.
09 00311'000433'.SPECL:SPECL
10 00312'000075'.CLR1:CLRL1
11 00313'044767 CHOCT:STA 1,YOCT
12 00314'030003-       LDA 2,OVLAP
13 00315'020021        LDA 0,21,0
14 00316'142400        SUB 2,0
15 00317'040021        STA 0,21,0
16                ;MAKE ADJUSTMENT FOR FACT THAT CTR1 HAS BEEN
17                ; INCREMENTED (OVLAP-2) EXTRA TIMES
18                ; THIS DIFFERS FROM LOOR1 OVERLAP LOGIC, IN THAT
19                ; CONTROL PASSES TO THE SAME PLACE (SAMY2) WHETHER
20                ; VALUES ARE OVERLAPPED OR WITHIN AN OCTANT.
21 00320'020747        LDA 0,R2
22 00321'112400        SUB 0,2
23 00322'024751        LDA 1,CTR1
24 00323'146400        SUB 2,1
25 00324'044747        STA 1,CTR1
26 00325'000401        JMP .+1
27 00326'040003-       STA 0,OVLAP
28 00327'002743        JMP @.CLR1
29                ;  THE FOLLOWING LOGIC ELIMINATES SEARCHING FOR
30                ;  LEAKERS IN THE BIT MATRIX IF THE OCTANT>0 AND Y=0
31 00330'030740 CONT2:LDA 2,MASK4
32 00331'113400        AND 0,2
33 00332'024750        LDA 1,YOCT
34 00333'125005        MOV 1,1,SNR
35 00334'000404        JMP CONT3
36 00335'151004        MOV 2,2,SZR
37 00336'000402        JMP CONT3
38 00337'000445        JMP END2    ; END OF OCTANT>0,Y=0 LOGIC
39 00340'040746 CONT3:STA 0,RRVY2
40 00341'151400        INC 2,2
41 00342'050737        STA 2,Y2
42 00343'175400 SAMY2:INC 3,3
43 00344'024722        LDA 1,MASK2
44 00345'167400        AND 3,1
45 00346'044736        STA 1,BITX2
46 00347'024716        LDA 1,MASK1
47 00350'167520        ANDZL 3,1
48 00351'125100        MOVL 1,1
49 00352'125100        MOVL 1,1
50 00353'125100        MOVL 1,1
51 00354'125300        MOVS 1,1
52                 ; AC1 CONTAINS X2 COORDINATE
53 00355'000401        JMP .+1
54 00356'034704        LDA 3,STRAD    ; AC3 CONTAINS ADDRESS OF STAR TABLE
55 00357'030722        LDA 2,Y2     ; AC2 CONTAINS STAR TABLE OFFSET BY Y2 COORDINATE
56 00360'157000        ADD 2,3
57 00361'031400        LDA 2,0,3
58 00362'133000        ADD 1,2
59 00363'021000        LDA 0,0,2
```

```
  0006  CNCLS
01 00364'034711        LDA 3,PT2AD
02 00365'000401        JMP .+1
03 00366'030716       LDA 2,BITX2
04 00367'157000        ADD 2,3
05 00370'031400        LDA 2,0,3
06 00371'113404        AND 0,2.SZR
07 00372'000412        JMP END2
08 00373'034021        LDA 3,21,0  ; FIND ADDRESS IN AUTO INCREMENTING
09                                 ; LOCATION 21
10 00374'025777        LDA 1,-1,3    ; TAKE ORIGINAL X,  PUT IN LEAKER LIST
11 00375'046022        STA 1,@22,0
12 00376'000401        JMP .+1
13 00377'025400        LDA 1,0,3    ; TAKE ORIGINAL Y - PUT IN LEAKER LIST
14 00400'046022        STA 1,@22,0
15 00401'010000-       ISZ CTLKR
16 00402'000402        JMP .+2
17 00403'000551        JMP LEKON
18 00404'010647 END2:  ISZ CTR1
19 00405'000402        JMP .+2
20 00406'000400        JMP .   ; ODD # OF WORDS
21 00407'010664        ISZ CTR1
22 00410'000617        JMP LOOP2
23 00411'020000-       LDA 0,CTLKR
24 00412'024537        LDA 1,P100
25 00413'123000        ADD 1,0
26 00414'040000-       STA 0,CTLKR
27 00415'002001-       JMP @SAVE
28 00416'000147'.LOP1:LOOP1
29 00417'001760 MSK1:  1760
30 00420'001061'STRA1:STARS
31 00421'005323'PT1A1:PATN1
32 00422'005363'WHR64:JMP64
33 00423'005403'WHR65:JMP65
34 00424'000271'.CTR: CTR
35 00425'000003-.OVLAP:OVLAP
36 00426'000227'.LOP2:LOOP2
37 00427'000002 T2:   2
38 00430'000017 MSK2:   17
39 00431'000303'.BTX1:BITX1
40 00432'000000 X1S:   0
41 00433'054777 SPECL:STA 3,X1S
42 00434'176520        SUBZL 3,3    ; +1
43 00435'173004        ADD 3,2,SZR
44 00436'000475        JMP COCT1
45 00437'034631        LDA 3,MASK4
46 00440'117404        AND 0,3.SZR
47 00441'000434        JMP CHCK1
48 00442'024003-       LDA 1,OVLAP
49 00443'125400        INC 1,1
50 00444'125400        INC 1,1
51 00445'044003-       STA 1.OVLAP
52 00446'020642        LDA 0,P32
53 00447'034743        LDA 3,X1S
54 00450'024616        LDA 1,MASK2
55 00451'167400        AND 3,1
56 00452'044631        STA 1,BITX1
57 00453'024612        LDA 1,MASK1
58 00454'167520        ANDZL 3,1
59 00455'125100        MOVL 1,1
```

```
  0007  CNCLS
01 00456'125100        MOVL 1,1
02 00457'125100        MOVL 1,1
03 00460'125300        MOVS 1,1
04 00461'034737        LDA 3,STRA1    ;ADDRESS OF STAR TABLE
05 00462'117000        ADD 0,3
06 00463'021400        LDA 0,0,3
07 00464'123000        ADD 1,0    ; AC0  CONTAINS X,Y ADDRESS IN MATRIX
08 00465'024607        LDA 1,PT1A0
09 00466'030615        LDA 2,ATX1
10 00467'034733        LDA 3,MMR60
11 00470'157000        ADD 2,3
12 00471'007400        JSR @0,3
13 00472'012732        ISZ @.CTR
14 00473'000401        JMP .+1
15 00474'002722        JMP @.LOP1
16 00475'152000 CHCK1:ADC 2,2    ; -1
17 00476'157004        ADD 2,3,SZR
18 00477'000434        JMP COCT1
19 00500'024003-       LDA 1,OVLAP
20 00501'125400        INC 1,1
21 00502'125400        INC 1,1
22 00503'044003-       STA 1,OVLAP
23 00504'020004-       LDA 0,P33
24 00505'034725        LDA 3,X1S
25 00506'024722        LDA 1,MSK2
26 00507'167400        AND 3,1
27 00510'046721        STA 1,@.ATX1
28 00511'024706        LDA 1,MSK1
29 00512'167520        ANDZL 3,1
30 00513'125100        MOVL 1,1
31 00514'125100        MOVL 1,1
32 00515'125100        MOVL 1,1
33 00516'125300        MOVS 1,1
34 00517'034701        LDA 3,STRA1    ; ADDRESS OF STAR TABLE
35 00520'117000        ADD 0,3    ; ADD Y1
36 00521'021400        LDA 0,0,3    ; GET ADDRESS OF THAT PART OF MATRIX
37 00522'123000        ADD 1,0    ; ADD X - AC0 CONTAINS XY ADDRESS IN MATRIX
38 00523'024676         LDA 1,PT1A1
39 00524'032705        LDA 2,@.ATX1
40 00525'034676        LDA 3,MMR65
41 00526'157000        ADD 2,3
42 00527'007400        JSR @0,3
43 00530'012674        ISZ @.CTR
44 00531'000401        JMP .+1
45 00532'002664        JMP @.LOP1
46 00533'032673 COCT1:LDA 2,@.OVLAP  ; DON'T STORE XOCT - LEAVE
47 00534'020020        LDA 0,20,0     ; FOR CANCELLATION WITH TABLE 2
48 00535'142400        SUB 2,0
49 00536'040020        STA 0,20,0
50                ;  MAKE ADJUSTMENT FOR FACT THAT CTR HAS BEEN
51                ;  INCREMENTED (OVLAP-2)/2 EXTRA TIMES
52 00537'020670        LDA 0,I2
53 00540'112400        SUB 0,2
54 00541'151220        MOVZR 2,2
55 00542'026662        LDA 1,@.CTR
56 00543'146400        SUB 2,1
57 00544'046660        STA 1,@.CTR
58 00545'000401        JMP .+1
59 00546'042657         STA 0,@.OVLAP
```

```
  000A  CNCLS
01 00547'000401          JMP .+1
02 00550'002656          JMP @.LOP2
03 00551'000144 9100: 100.
04 00552'177755 M19:-19.
05 00553'177767 M9:-9.
06 00554'020775 LEKON:LDA 0,P100  ; WHEN 100 LEAKERS ARE ENCOUNTERED,
07 00555'040000-         STA 0,CTLK9  ; TAKE 91ST PAIR'S X & Y VALUES
08 00556'030022          LDA 2,22,0   ; AND STORE THE Y VALUE
09 00557'020773          LDA 0,M19    ; IN THE REMAINING Y LEAKER WORDS,
10 00560'113000          ADD 0,2      ; & X+2 IN EACH OF THE REMAINING
11 00561'035000          LDA 3,0,2    ; X LEAKER WORDS.
12 00562'175400          INC 3,3   ;  THUS DISPLAY WITH HORIZONTAL
13 00563'175400          INC 3,3   ; ENDING LINE INDICATES
14 00564'151400          INC 2,2   ; AN OVERFLOW OF LEAKERS
15 00565'021000          LDA 0,0,2
16 00566'151400          INC 2,2
17 00567'024764          LDA 1,M9
18 00570'055000 LKLUP:STA 3,0,2  ; AC3 CONTAINS X VALUE
19 00571'175400          INC 3,3     ; AC0 CONTAINS Y VALUE
20 00572'175400          INC 3,3     ; AC1 CONTAINS COUNTER
21 00573'151400          INC 2,2     ; AC2 CONTAINS LEAKER ADDRESS
22 00574'041000          STA 0,0,2
23 00575'151400          INC 2,2
24 00576'125404          INC 1,1,SZR
25 00577'000771          JMP LKLUP
26 00600'126520          SUBZL 1,1
27 00601'044002-         STA 1,LKOVF   ; SET LEAKER OVERFLOW INDICATOR
28 00602'000401          JMP .+1
29 00603'002001-         JMP @SAVE
30 00604'050540 BITS: STA 2,OFFST   ; VALUE OF BITx
31 00605'054536          STA 3,SAVER
32 00606'131000          MOV 1,2   ; 2 NOW CONTAINS ADDRESS OF PATN1
33 00607'115000          MOV 0,3   ; 3 CONTAINS ADDRESS IN BIT MATRIX
34 00610'020534          LDA 0,OFFST
35 00611'113000          ADD 0,2
36 00612'021000          LDA 0,0,2
37 00613'025400          LDA 1,0,3
38 00614'100000          COM 0,0
39 00615'107400          AND 0,1
40 00616'106000          ADC 0,1
41 00617'045400          STA 1,0,3
42 00620'021000          LDA 0,0,2
43 00621'025500          LDA 1,100,3
44 00622'100000          COM 0,0
45 00623'107400          AND 0,1
46 00624'106000          ADC 0,1
47 00625'045500          STA 1,100,3
48 00626'021000          LDA 0,0,2
49 00627'175400          INC 3,3
50 00630'025577          LDA 1,177,3
51 00631'100000          COM 0,0
52 00632'107400          AND 0,1
53 00633'106000          ADC 0,1
54 00634'045577          STA 1,177,3
55 00635'002506          JMP @SAVER
56 00636'054505 BIT16:STA 3,SAVER
57 00637'030504          LDA 3,SEC16
58 00640'000403          JMP .+3
59 00641'054502 BIT17:STA 3,SAVER
```

```
  0009  CNCLS
01 00642'034504          LDA 3,SEC17
02 00643'054504          STA 3,SECPT
03 00644'050500          STA 2,OFFST
04 00645'131000          MOV 1,2
05 00646'115000          MOV 0,3
06 00647'020475          LDA 0,OFFST
07 00650'113000          ADD 0,2
08 00651'021000          LDA 0,0,2
09 00652'025400          LDA 1,0,3
10 00653'100000          COM 0,0
11 00654'107400          AND 0,1
12 00655'106000          ADC 0,1
13 00656'045400          STA 1,0,3
14 00657'020470          LDA 0,SECP1   ; SECOND BIT PATTERN WORD
15 00660'025401          LDA 1,1,3
16 00661'100000          COM 0,0
17 00662'107400          AND 0,1
18 00663'106000          ADC 0,1
19 00664'045401          STA 1,1,3
20 00665'021000          LDA 0,0,2
21 00666'025500          LDA 1,100,3
22 00667'100000          COM 0,0
23 00670'107400          AND 0,1
24 00671'106000          ADC 0,1
25 00672'045500          STA 1,100,3
26 00673'020454          LDA 0,SECPT
27 00674'025501          LDA 1,101,3
28 00675'100000          COM 0,0
29 00676'107400          AND 0,1
30 00677'106000          ADC 0,1
31 00700'045501          STA 1,101,3
32 00701'021000          LDA 0,0,2
33 00702'175400          INC 3,3
34 00703'025577          LDA 1,177,3
35 00704'100000          COM 0,0
36 00705'107400          AND 0,1
37 00706'106000          ADC 0,1
38 00707'045577          STA 1,177,3
39 00710'020437          LDA 0,SECPT
40 00711'175400          INC 3,3
41 00712'025577          LDA 1,177,3
42 00713'100000          COM 0,0
43 00714'107400          AND 0,1
44 00715'106000          ADC 0,1
45 00716'045577          STA 1,177,3
46 00717'002424          JMP @SAVER
47                ;   BIT PATTERN ASSIGNMENT ROUTINES FOR Y=64 IN OVERLAP
48 00720'050424 R64S:    STA 2,OFFST   ; VALUE OF BITX
49 00721'054422          STA 3,SAVER
50 00722'131000          MOV 1,2    ; 2 NOW CONTAINS ADDRESS OF PATN1
51 00723'115000          MOV 0,3    ; 3 CONTAINS ADDRESS IN BIT MATRIX
52 00724'020420          LDA 0,OFFST
53 00725'113000          ADD 0,2
54 00726'021000          LDA 0,0,2
55 00727'025400          LDA 1,0,3
56 00730'100000          COM 0,0
57 00731'107400          AND 0,1
58 00732'106000          ADC 0,1
59 00733'045400          STA 1,0,3
```

```
  0010  CNCLS
01 00734'021000        LDA 0,0,2
02 00735'025500        LDA 1,100,3
03 00736'100000        COM 0,0
04 00737'107400        AND 0,1
05 00740'106000        ADC 0,1
06 00741'045500        STA 1,100,3
07 00742'002401        JMP @SAVER
08 00743'000000 SAVER:0
09 00744'000000 OFFST:0
10 00745'100000 SEC16:100000
11 00746'140000 SEC17:140000
12 00747'000000 SECPT:0
13 00750'054773 R6416:STA 3,SAVER
14 00751'034774        LDA 3,SEC16
15 00752'000403        JMP .+3
16 00753'054770 R6417:STA 3,SAVER
17 00754'034772        LDA 3,SEC17
18 00755'054772        STA 3,SECPT
19 00756'050766        STA 2,OFFST
20 00757'131000        MOV 1,2    ; AC2 CONTAINS ADDRESS OF PATN1
21 00760'115000        MOV 0,3    ; AC3 CONTAINS ADDRESS IN BIT MATRIX
22 00761'020763        LDA 0,OFFST
23 00762'113000        ADD 0,2
24 00763'021000        LDA 0,0,2
25 00764'025400        LDA 1,0,3
26 00765'100000        COM 0,0
27 00766'107400        AND 0,1
28 00767'106000        ADC 0,1
29 00770'045400        STA 1,0,3
30 00771'020756        LDA 0,SECPT     ; SECOND BIT PATTERN WORD
31 00772'025401        LDA 1,1,3
32 00773'100000        COM 0,0
33 00774'107400        AND 0,1
34 00775'106000        ADC 0,1
35 00776'045401        STA 1,1,3
36 00777'021000        LDA 0,0,2
37 01000'025500        LDA 1,100,3
38 01001'100000        COM 0,0
39 01002'107400        AND 0,1
40 01003'106000        ADC 0,1
41 01004'045500        STA 1,100,3
42 01005'020742        LDA 0,SECPT
43 01006'025501        LDA 1,101,3
44 01007'100000        COM 0,0
45 01010'107400        AND 0,1
46 01011'106000        ADC 0,1
47 01012'045501        STA 1,101,3
48 01013'002730        JMP @SAVER
49                ;  BIT PATTERN ASSIGNMENT ROUTINES FOR Y=65 IN OVERLAP
50 01014'050730 R65S: STA 2,OFFST   ;VALUE OF BITX
51 01015'054726        STA 3,SAVER
52 01016'131000        MOV 1,2
53                   ;  AC2 CONTAINS ADDRESS OF PATN1
54 01017'115000        MOV 0,3    ; AC3 CONTAINS ADDRESS IN BIT MATRIX
55 01020'020724        LDA 0,OFFST
56 01021'113000        ADD 0,2
57 01022'021000        LDA 0,0,2
58 01023'025400        LDA 1,0,3
59 01024'100000        COM 0,0
```

```
  0011  CNCLS
01 01025'107400          AND 0,1
02 01026'106000          ADC 0,1
03 01027'045400          STA 1,0,3
04 01030'002713          JMP @SAVER
05 01031'054712 B6516:STA 3,SAVER
06 01032'034713          LDA 3,SEC16
07 01033'000403          JMP .+3
08 01034'054707 B6517: STA 3,SAVER
09 01035'034711          LDA 3,SEC17
10 01036'054711          STA 3,SECPT
11 01037'050705          STA 2,OFFST
12 01040'131000          MOV 1,2    ; AC2 CONTAINS PATN1 ADDRESS
13 01041'115000          MOV 0,3    ; AC3 CONTAINS ADDRESS IN BIT MATRIX
14 01042'020702          LDA 0,OFFST
15 01043'113000          ADD 0,2
16 01044'021000          LDA 0,0,2
17 01045'025400          LDA 1,0,3
18 01046'100000          COM 0,0
19 01047'107400          AND 0,1
20 01050'106000          ADC 0,1
21 01051'045400          STA 1,0,3
22 01052'020675          LDA 0,SECPT
23 01053'025401          LDA 1,1,3
24 01054'100000          COM 0,0
25 01055'107400          AND 0,1
26 01056'106000          ADC 0,1
27 01057'045401          STA 1,1,3
28 01060'002663          JMP @SAVER
29 01061'001123'STARS:ROW1
30 01062'001223'         ROW2
31 01063'001323'         ROW3
32 01064'001423'         ROW4
33 01065'001523'         ROW5
34 01066'001623'         ROW6
35 01067'001723'         ROW7
36 01070'002023'         ROW8
37 01071'002123'         ROW9
38 01072'002223'         ROW10
39 01073'002323'         ROW11
40 01074'002423'         ROW12
41 01075'002523'        ROW13
42 01076'002623'         ROW14
43 01077'002723'         ROW15
44 01100'003023'         ROW16
45 01101'003123'         ROW17
46 01102'003223'         ROW18
47 01103'003323'         ROW19
48 01104'003423'         ROW20
49 01105'003523'         ROW21
50 01106'003623'         ROW22
51 01107'003723'         ROW23
52 01110'004023'         ROW24
53 01111'004123'         ROW25
54 01112'004223'         ROW26
55 01113'004323'         ROW27
56 01114'004423'         ROW28
57 01115'004523'         ROW29
58 01116'004623'         ROW30
59 01117'004723'         ROW31
```

```
  0012  CNCL5
01 01120'005023'        ROW32
02 01121'005123'        ROW33
03 01122'005223'        ROW34
04        000100 ROW1:  .BLK 100
05        000100 ROW2:  .BLK 100
06        000100 ROW3:  .BLK 100
07        000100 ROW4:  .BLK 100
08        000100 ROW5:  .BLK 100
09        000100 ROW6:  .BLK 100
10        000100 ROW7:  .BLK 100
11        000100 ROW8:  .BLK 100
12        000100 ROW9:  .BLK 100
13        000100 ROW10: .BLK 100
14        000100 ROW11: .BLK 100
15        000100 ROW12: .BLK 100
16        000100 ROW13: .BLK 100
17        000100 ROW14: .BLK 100
18        000100 ROW15: .BLK 100
19        000100 ROW16: .BLK 100
20        000100 ROW17: .BLK 100
21        000100 ROW18: .BLK 100
22        000100 ROW19: .BLK 100
23        000100 ROW20: .BLK 100
24        000100 ROW21: .BLK 100
25        000100 ROW22: .BLK 100
26        000100 ROW23: .BLK 100
27        000100 ROW24: .BLK 100
28        000100 ROW25: .BLK 100
29        000100 ROW26: .BLK 100
30        000100 ROW27: .BLK 100
31        000100 ROW28: .BLK 100
32        000100 ROW29: .BLK 100
33        000100 ROW30: .BLK 100
34        000100 ROW31: .BLK 100
35        000100 ROW32: .BLK 100
36        000100 ROW33: .BLK 100
37        000100 ROW34: .BLK 100
38 05323'160000 RATN1:160000
39 05324'070000        070000
40 05325'034000        034000
41 05326'016000        016000
42 05327'007000        007000
43 05330'003400        003400
44 05331'001600        001600
45 05332'000700         000700
46 05333'000340        000340
47 05334'000160        00160
48 05335'000070        000070
49 05336'000034        000034
50 05337'000016        000016
51 05340'000007        000007
52 05341'000003        000003
53 05342'000001        000001
54 05343'000604'JMRLS:BITS
55 05344'000604'        BITS
56 05345'000604'        BITS
57 05346'000604'        BITS
58 05347'000604'        BITS
59 05350'000604'        BITS
```

```
  0013  CNCLS
01 05351'000604'       BITS
02 05352'000604'       BITS
03 05353'000604'       BITS
04 05354'000604'       BITS
05 05355'000604'       BITS
06 05356'000604'       BITS
07 05357'000604'       BITS
08 05360'000604'       BITS
09 05361'000636'       BIT16
10 05362'000641'       BIT17
11 05363'000720'JMP64:B64S
12 05364'000720'       B64S
13 05365'000720'       B64S
14 05366'000720'       B64S
15 05367'000720'       B64S
16 05370'000720'      B64S
17 05371'000720'       B64S
18 05372'000720'       B64S
19 05373'000720'       B64S
20 05374'000720'       B64S
21 05375'000720'       B64S
22 05376'000720'       B64S
23 05377'000720'       B64S
24 05400'000720'       B64S
25 05401'000750'       B6416
26 05402'000753'       B6417
27 05403'001014'JMP65:B65S
28 05404'001014'       B65S
29 05405'001014'       B65S
30 05406'001014'       B65S
31 05407'001014'       B65S
32 05410'001014'       B65S
33 05411'001014'       B65S
34 05412'001014'       B65S
35 05413'001014'       B65S
36 05414'001014'        B65S
37 05415'001014'       B65S
38 05416'001014'       B65S
39 05417'001014'       B65S
40 05420'001014'       B65S
41 05421'001031'       B6516
42 05422'001031'       B6517
43 05423'100000 PATN2:100000
44 05424'040000        040000
45 05425'020000        020000
46 05426'010000        010000
47 05427'004000        004000
48 05430'002000        002000
49 05431'001000        001000
50 05432'000400        000400
51 05433'000200        000200
52 05434'000100        000100
53 05435'000040        000040
54 05436'000020        000020
55 05437'000010        000010
56 05440'000004        000004
57 05441'000002        000002
58 05442'000001        000001
59 05443'100001 DTCDE:-32767.


  0014  CNCLS
01 05444'000000        0
02        000312 LK1:  .BLK 312
03                     .END   ;  END OF CNCLSTR
```

```
0015  CNCLS

B6416   000750'    10/13   13/25
B6417   000753'    10/16   13/26
B645    000720'     9/48   13/11   13/12   13/13   13/14   13/15   13/16   13/17
                   13/18   13/19   13/20   13/21   13/22   13/23   13/24
B6516   001031'    11/05   13/41
B6517   001034'    11/08   13/42
B658    001014'    10/50   13/27   13/28   13/29   13/30   13/31   13/32   13/33
                   13/34   13/35   13/36   13/37   13/38   13/39   13/40
BIT16   000636'     8/56   13/09
BIT17   000641'     8/59   13/10
BITS    000604'     8/30   12/54   12/55   12/56   12/57   12/58   12/59   13/01
                   13/02   13/03   13/04   13/05   13/06   13/07   13/08
BITX1   000303'     3/39    3/56    5/03    6/39    6/56    7/09
BITX2   000304'     5/04    5/45    6/03
CHCK1   000475'     6/47    7/16
CHOCT   000313'     4/34    4/37    5/11
CLRL1   000075'     2/11    2/29    5/10
CNCLS   000021'     1/32
COCT1   000533'     6/44    7/18    7/46
CONT1   000165'     3/24    3/33
CONT2   000330'     4/22    5/31
CONT3   000340'     5/35    5/37    5/39
CTLKR   000000-     1/09    1/50    2/05    2/15    2/20    2/24    6/15    6/23
                    6/26    8/07
CTR     000271'     1/28    4/04    4/07    4/52    6/34
CTR1    000273'     1/29    4/54    5/23    5/25    6/18    6/21
CTRC    000006'     1/21    2/12    2/18    2/30    3/10
CTTR1   000002'     1/17    2/06
CTTR2   000003'     1/18    2/01
DTCDE   005443'    13/59
END2    000404'     5/38    6/07    6/18
I2      000427'     6/37    7/52
JMP44   005363'     6/32   13/11
JMP45   005403'     6/33   13/27
JMPLS   005343'     4/53   12/54
LEKON   000554'     1/30    6/17    8/06
LK1     005445'     1/19   14/02
LK1AD   000004'     1/19    1/42
LKLUP   000570'     8/18    8/25
LKOVF   000002-     1/11    1/51    8/27
LOOP    000102'     2/34    3/11
LOOP1   000147'     3/12    4/08    6/28
LOOP2   000227'     4/10    6/22    6/36
M100    000020'     1/31    2/04    2/21
M19     000552'     8/04    8/09
M64     000005'     1/20    2/29
M9      000553'     8/05    8/17
MASK1   000265'     3/40    4/48    5/46    6/57
MASK2   000266'     3/37    4/49    5/43    6/54
MASK3   000267'     3/17    4/15    4/50
MASK4   000270'     3/34    4/35    4/51    5/31    6/45
MSK1    000417'     6/29    7/28
MSK2    000430'     6/38    7/25
OFFST   000744'     8/30    8/34    9/03    9/06    9/48    9/52   10/09   10/19
                   10/22   10/50   10/55   11/11   11/14
OVLAP   000003-     1/12    1/27    4/38    4/41    5/12    5/27    6/35    6/48
                    6/51    7/19    7/22
P100    000551'     6/24    8/03    8/06
P2      000307'     5/07    5/21
```

0016 CNCLS

P32	000310'	5/08	6/52						
P33	000004-	1/13	4/42	7/23					
P34	000007'	1/22	2/31						
PATN1	005323'	4/55	6/31	12/38					
PATN2	005423'	4/56	13/43						
PRVY1	000305'	1/24	3/14	3/33	5/05				
PRVY2	000306'	1/25	4/12	5/06	5/39				
PT1A1	000421'	6/31	7/38						
PT1AD	000274'	3/55	4/55	7/08					
PT2AD	000275'	4/56	6/01						
ROW1	001123'	4/46	11/29	12/04					
ROW10	002223'	11/38	12/13						
ROW11	002323'	11/39	12/14						
ROW12	002423'	11/40	12/15						
ROW13	002523'	11/41	12/16						
ROW14	002623'	11/42	12/17						
ROW15	002723'	11/43	12/18						
ROW16	003023'	11/44	12/19						
ROW17	003123'	11/45	12/20						
ROW18	003223'	11/46	12/21						
ROW19	003323'	11/47	12/22						
ROW2	001223'	11/30	12/05						
ROW20	003423'	11/48	12/23						
ROW21	003523'	11/49	12/24						
ROW22	003623'	11/50	12/25						
ROW23	003723'	11/51	12/26						
ROW24	004023'	11/52	12/27						
ROW25	004123'	11/53	12/28						
ROW26	004223'	11/54	12/29						
ROW27	004323'	11/55	12/30						
ROW28	004423'	11/56	12/31						
ROW29	004523'	11/57	12/32						
ROW3	001323'	11/31	12/06						
ROW30	004623'	11/58	12/33						
ROW31	004723'	11/59	12/34						
ROW32	005023'	12/01	12/35						
ROW33	005123'	12/02	12/36						
ROW34	005223'	12/03	12/37						
ROW4	001423'	11/32	12/07						
ROW5	001523'	11/33	12/08						
ROW6	001623'	11/34	12/09						
ROW7	001723'	11/35	12/10						
ROW8	002023'	11/36	12/11						
ROW9	002123'	11/37	12/12						
ROWAD	000263'	2/33	4/46						
SAMY1	000171'	3/16	3/37						
SAMY2	000343'	4/14	4/45	5/42					
SAVE	000001-	1/10	1/32	6/27	R/29				
SAVER	000743'	R/31	R/55	R/56	R/5R	9/46	9/49	10/07	10/08
		10/13	10/16	10/48	10/51	11/04	11/05	11/08	11/28
SEC16	000745'	R/57	10/10	10/14	11/06				
SEC17	000746'	9/01	10/11	10/17	11/09				
SECRT	000747'	9/02	9/14	R/26	R/3R	10/12	10/18	10/30	10/42
		11/10	11/22						
SPECL	000433'	5/08	6/41						
STARS	001061'	4/47	6/30	11/29					
STRA1	000420'	6/30	7/04	7/34					
STRAD	000264'	3/48	4/47	5/54					
TB1AD	000000'	1/15	1/34						

```
  0017  CNCLS

TH2AD   000001'     1/16    1/38
TRNER   000057'     2/08    2/12
WHERE   000272'     3/57    4/53
WHR64   000422'     6/32    7/10
WHR65   000423'     6/33    7/40
X1      000276'     4/57
X1S     000432'     6/40    6/41    6/53    7/24
X2      000300'     4/31    4/44    4/59
Y1      000277'     3/36    3/49    4/58
Y2      000301'     4/43    5/01    5/41    5/55
YOCT    000302'     1/23    3/22    4/20    5/02    5/11    5/33
.BTX1   000431'     6/39    7/27    7/39
.CLR1   000312'     5/10    5/28
.CTR    000424'     6/34    7/13    7/43    7/55    7/57
.CTR1   000016'     1/29    2/10
.CTRR   000015'     1/28    2/09
.LKDN   000017'     1/30    2/17
.LOP1   000416'     6/28    7/15    7/45
.LOP2   000426'     6/36    8/02
.OVL1   000014'     1/27    1/53
.OVLA   000425'     6/35    7/46    7/59
.P2     000013'     1/26    1/52
.PRY1   000011'     1/24    1/48
.PRY2   000012'     1/25    1/49
.SPEC   000311'     3/32    5/09
.YOCT   000010'     1/23    1/47
```

```
  0001  TRKIN            TRKIN15A
01                      .TITL TRKIN
02                      .ENT TRKIN
03                      .EXTN TRNSV,RCVSV
04                      .EXTN LSXYZ
05                ;*****************************************************************
06                ;*****************************************************************
07                ;  TRACK INITIATION PROGRAM 4/22/75
08                ;       THIS PROGRAM RECEIVES 3 FRAMES OF DATA FROM ALL FOUR CAMS,
09                ;       AND USING RELATIVE FRAME 2 AS THE BASIS, DETERMINES IF THE
10                ;       AVERAGE OF A GIVEN SET OF COORDINATE PAIRS FROM FRAMES 1 & 3
11                ;       MATCH A COORDINATE PAIR IN FRAME 2, THUS ESTABLISHING A
12                ;       TRACK.
13                ;
14                ;       WHEN SUCH A MATCH OCCURS, THE COORDINATE PAIRS FROM FRAME 1
15                ;       AND FRAME 3 ARE STORED IN A LIST, AND AFTER PROCESSING ALL
16                ;       POINTS FOR A GIVEN 3 FRAMES, THIS LIST IS TRANSMITTED TO
17                ;       RECM2  ( RECONSTRUCTION MINI 2).
18                ;*****************************************************************
19                ;*****************************************************************
20                      .ZREL
21        000004 LSVAR: .BLK 4    ; THIS BLOCK WILL BE USED TO PASS 4 VARIABLES
22                                ; TO THE LSXYZ ROUTINE.
23                                ;  1) .R - POINTER FOR TRACK INITIATION DATA
24                                ;  2) .Y - POINTER FOR Y DATA AREA
25                                ;  3) TICTR - NEGATIVE WORD COUNT OF DATA AT .R
26                                ;  4) CTRY  - NEGATIVE WORD COUNT OF DATA AT .Y
27 00004-000045-.READY:READY
28 00005-170000 HCAMK:170000
29 00006-100000 TIMCA:100000
30 00007-000004 TITYP:4
31 00010-000404 .NMAX:404
32 00011-001450 IDATA:1450  ; INCREMENT OF DATA BLOCKS, FIRST TWO WORDS OF
33                          ; WHICH WILL BE -32767. & 0.
34 00012-000100 IHDR:100    ; INCREMENT OF HEADER BLOCKS
35 00013-002000 ITID:2000   ; INCREMENT OF TRACK INITIATION DATA BLOCK,
36                          ; FIRST TWO WORDS OF WHICH WILL BE -32767. & 0.
37 00014-000000 .A:0        ; POINTER - AREA A'S DATA
38 00015-000000 .B:0        ; POINTER - AREA B'S DATA
39 00016-000000 .C:0        ; POINTER - AREA C'S DATA
40 00017-000000 .D:0        ; POINTER - AREA D'S DATA
41 00020-000000 .HA:0       ; POINTER - AREA A'S HEADERS
42 00021-000000 .HB:0       ; POINTER - AREA B'S HEADERS
43 00022-000000 .HC:0       ; POINTER - AREA C'S HEADERS
44 00023-000000 .HD:0       ; POINTER - AREA D'S HEADERS
45 00024-000000 .MTRX:0    ; POINTER - BIT MATRIX
46 00025-000312 ICAMD:312   ; INCREMENT OF INDIVIDUAL CAM DATA
47 00026-000020 ICAMH:20    ; INCREMENT OF INDIVIDUAL CAM HEADER
48 00027-000000 .CURH:0     ; CURRENT HEADER POINTER - WITHIN LARGE AREA
49 00030-000000 .CURD:0     ; CURRENT DATA POINTER - WITHIN LARGE AREA
50 00031-000000 .F:0        ; POINTER - FILLING DATA AREA
51 00032-000000 .X:0        ;POINTER - X DATA AREA
52 00033-000000 .Y:0        ; POINTER - Y DATA AREA
53 00034-000000 .Z:0        ; POINTER - Z DATA AREA
54 00035-000000 .W:0        ; POINTER - WRAP AROUND DATA AREA
55 00036-000000 .HF:0       ; POINTER - FILLING HEADER AREA
56 00037-000000 .HX:0       ; POINTER - X HEADER AREA
57 00040-000000 .HY:0       ; POINTER - Y HEADER AREA
58 00041-000000 .HZ:0       ; POINTER - Z HEADER AREA
59 00042-000000 .HW:0       ; POINTER - WRAPAROUND HEADER AREA
```

```
  0002  TRKIN
01 00043-000000 FULL:0          ; WHEN FULL = 0, DATA IS BEING READ INTO FILLING BUFFER
02                              ; WHEN FULL = 1, BUFFER IS FULL, I.E. DATA HAS
03                              ;                HEEN RECEIVED FROM ALL 4 CAMS.
04 00044-000000 RSTIN:0         ; WHEN RSTIN = 0, CONTINUE NORMALLY
05                              ; WHEN RSTIN = 1, RETURN TO RESET AREA & RESTART THE PROGRAM.
06 00045-100003 READY:-32765.   ; READY ACKNOWLEDGE BLOCK
07 00046-000000         0       ;TI MCA ADDRESS - TO BE SET
08 00047-000004         4       ; TI PROGRAM TYPE
09 00050-000000         0       ; 0 INDICATES ACCEPT, 1 INDICATES REJECT
10 00051-000000         0       ; UNUSED
11 00052-020000 CAMAC:020000    ; LISTING OF CAM MCA ADDRESSES
12 00053-040000         040000  ; FROM WHICH THIS PROGRAM CAN RECEIVE.
13 00054-030000         030000  ; ADDRESSES ESTABLISHED AT TIME OF POLLING.
14 00055-060000         060000
15 00056-000000 CAMWK:0     ; WORKING CAM MCA ADDRESS LIST.
16 00057-000000         0   ; EACH TIME THE FILL PROCESS IS STARTED,
17 00060-000000         0   ; THIS LIST IS INITIALIZED TO VALUES OF CAMAC.
18 00061-000000         0
19 00062-000052-.CAMAC:CAMAC  ; ADDRESS OF ACCEPTABLE CAM LIST
20 00063-000056-.CAMWK:CAMWK  ; ADDRESS OF WORKING LIST
21 00064-000000 CAMCT:0
22 00065-120000 R1MCA:120000
23 00066-050000 R2MCA:050000
24 00067-000136'.TR1:TR1
25 00070-177776 M2:-2
26 00071-000000 CTR:0
27 00072-000000 TICTR:0
28 00073-000302'.TR2:TR2
29 00074-000314'.TR3:TR3
30 00075-000330'.TR4:TR4
31 00076-000000 RTRN:0
32 00077-177774 M4:-4
33 00100-000000 CTRI:0
34 00101-000000 CMCTR:0
35 00102-000453'INTAD:INTSV
36 00103-177767 INTMK:177767
37 00104-000000 SAVAC:0    ; AC0
38 00105-000000         0   ;AC1
39 00106-000000         0   ; AC2
40 00107-000000         0   ; AC3
41 00110-000000         0   ; CARRY BIT
42 00111-000000 SAVIN:0
43 00112-000007 SEVEN:7
44 00113-000000 STATP:0
45 00114-000010 TIMOT:1412
46 00115-000002 CNTOT:1814
47 00116-000001 CNTOR:1815
48 00117-000000 RWDAD:0
49 00120-000000 RWDCT:0
50       000020 PSTHD: .BLK 20   ;RESET HEADER
51 00141-000121-.PSTHD:RSTHD
52 00142-177760 M16:-16.
53 00143-100004 MDCDE:-32764.
54 00144-000413'.HUMP:HUMP
55 00145-000361'.INPTF:INPTF
56 00146-001055'.SETM:SETM
57 00147-000766'.SETHD:SETHD
58 00150-001205'.CMCKM:CMCKM
59 00151-177777 .TRNSV:TRNSV
```

```
  0003  TRKIN
01 00152-177777 .RCVSV:RCVSV
02 00153-000545'.TNGCM:TNGCM
03 00154-000A06'.TRDYC:TRDYC
04 00155-100001 DTCDE:-32767.
05 00156-000000 CTRF:0      ; CTR - SQUASHED F AREA
06 00157-000000 CTRX:0      ; CTR - SQUASHED X AREA
07 00160-000000 CTRY:0      ; CTR - SQUASHED Y AREA
08 00161-000000 CTRZ:0      ; CTR - SQUASHED Z AREA
09 00162-000166-CTRA0:CTRLS
10 00163-000000 CTRCT:0
11 00164-000172-SCMA0:SCMLS
12 00165-000000 SCMCT:0
13        000004 CTRL9: .BLK 4
14        000004 SCML5: .BLK 4
15 00176-000000 RTRNF:0
16 00177-000000 RTRNS:0
17 00200-000A7A'.SDSHF:SDSHF
18 00201-000000 RTRNH:0
19 00202-000000 RTRNM:0
20 00203-000000 RTRNC:0
21 00204-100000 MSKPR:100000
22 00205-177777 .LSXYZ:LSXYZ
23 00206-100000 PATRN:100000
24 00207-040000         040000
25 00210-020000         020000
26 00211-010000         010000
27 00212-004000         004000
28 00213-002000         002000
29 00214-001000         001000
30 00215-000400         000400
31 00216-000200         000200
32 00217-000100         000100
33 00220-000040         000040
34 00221-000020         000020
35 00222-000010         000010
36 00223-000004         000004
37 00224-000002         000002
38 00225-000001         000001
39 00226-000040 P32:32.
40 00227-177600 M128:-128.
41 00230-000000 CTRM:0
42 00231-007760 MSKL4:7760
43 00232-000206-.PATRN:PATRN
44 00233-000017 MKU12:17
45 00234-000360 MKB11:360
46 00235-000000 SVXA0:0
47 00236-000000 MCTR1:0
48 00237-000000 MCTR2:0
49 00240-000000 MTXCT:0
50 00241-000000 XCT:0
51 00242-000000 ZCT:0
52 00243-000000 YCT:0
53                ;*****************************************************
54                ;   FORMAT OF HEADER TO RECM2
55                ;       WORD 1 = HEADER CODE = -32764.
56                ;       WORD 2 = MCA ADDRESS OF SENDING COMPUTER
57                ;       WORD 3 = -WORD COUNT OF DATA WHICH FOLLOWS
58                ;       WORD 4 = 0, BECAUSE DATA BLOCK FOLLOWS
59                ;       WORD 5 = # OF LEAKERS FROM 1ST CAM IN X AREA
```

```
 0004  TRKIN
01                ;       WORD 6 = # OF LEAKERS FROM 1ST CAM IN Z AREA
02                ;       WORD 7 = # OF LEAKERS FROM 2ND CAM IN X AREA
03                ;       WORD 8 = # OF LEAKERS FROM 2ND CAM IN Z AREA
04                ;       WORD 9 = # OF LEAKERS FROM 3RD CAM IN X AREA
05                ;       WORD 10= # OF LEAKERS FROM 3RD CAM IN Z AREA
06                ;       WORD 11= # OF LEAKERS FROM 4TH CAM IN X AREA
07                ;       WORD 12= # OF LEAKERS IN 4TH CAM IN Z AREA
08                ;       WORD 13= MCA ADDRESSES INDICATING ORDER THAT
09                ;                CAMS' DATA ARE IN X AREA
10                ;                BITS 0-3 = 4TH CAM'S MCA ADDRESS
11                ;                BITS 4-7 = 3RD CAM'S MCA ADDRESS
12                ;                BITS 8-11= 2ND CAM'S MCA ADDRESS
13                ;                BITS 12-15=1ST CAM'S MCA ADDRESS
14                ;       WORD 14 = FRAME # FROM ALL X DATA
15                ;       WORD 15 = FRAME # FROM ALL Z DATA
16                ;       WORD 16 = -WORD COUNT OF X AREA DATA FROM ALL 4 CAMS
17                ;*****************************************************************
18        000020 R2HDR: .BLK 20  ; RECM2 HEADER
19 00264-000244-.R2HDR:R2HDR
20 00265-000000 M2CT:0
21 00266-000000 .R2:0     ; RDINTER - TRACK INITIATION DATA
22 00267-000004 R4:4
23 00270-177773 M5:-5.
24 00271-000000 SVMCA:0
25 00272-000000 NZCAM:0
26 00273-000000 ALCAM:0
27                .NREL
28 00000'102400 TRKIN:SUB 0,0     ; ESTABLISH LOCATIONS 0 & 1 FOR INTERRUPTS
29 00001'040000       STA 0,0,0
30 00002'020102-      LDA 0,INTAD
31 00003'040001       STA 0,1,0
32 00004'062677       DICC 0,CRU    ; IO RESET
33 00005'020103-      LDA 0,INTMK
34 00006'062077       DOB 0,CRU
35 00007'000401       JMP .+1
36 00010'034004-      LDA 3,.READY
37 00011'060207       NIOC MCAR
38 00012'062407       DIC 0,MCAR    ; FIND OUT MCA CODE FOR THIS MACHINE
39 00013'024005-      LDA 1,MCAMK
40 00014'107400       AND 0,1
41 00015'044006-      STA 1,TIMCA    ; STORE MCA CODE IN TRACK INITIATION VARIABLE &
42 00016'045401       STA 1,1,3      ; READY BLOCK
43                ;*****************************************************************
44                ;  LARGE BLOCKS OF STORAGE ARE ESTABLISHED OUTSIDE OF SAVE FILE AREA
45 00017'034010-      LDA 3,.NMAX   ; FIND OUT NMAX ADDRESS
46 00020'031400       LDA 2,0,3     ; ESTABLISH RDINTERS TO DATA & HEADER
47 00021'050014-      STA 2,.A      ; RDINTER - AREA A'S DATA
48 00022'024011-      LDA 1,IDATA
49 00023'133000       ADD 1,2
50 00024'050015-      STA 2,.B      ; RDINTER - AREA B'S DATA
51 00025'133000       ADD 1,2
52 00026'050016-      STA 2,.C      ; RDINTER - AREA C'S DATA
53 00027'133000       ADD 1,2
54 00030'050017-      STA 2,.D      ; RDINTER - AREA D'S DATA
55 00031'133000       ADD 1,2
56 00032'050020-      STA 2,.HA     ; RDINTER - AREA A'S HEADERS
57 00033'024012-      LDA 1,IHDR
58 00034'133000       ADD 1,2
59 00035'050021-      STA 2,.HB     ; RDINTER - AREA B'S HEADERS
```

```
  0005  TRKIN
01 00036'133000          ADD 1,2
02 00037'050022-         STA 2,.HC     ; POINTER - AREA C'S HEADERS
03 00040'133000          ADD 1,2
04 00041'050023-         STA 2,.HD     ; POINTER - AREA D'S HEADERS
05 00042'133000          ADD 1,2
06 00043'050266-         STA 2,.R2     ; POINTER - TRACK INITIATION DATA
07 00044'020155-         LDA 0,DTCDE
08 00045'041000          STA 0,0,2
09 00046'151400          INC 2,2
10 00047'102400          SUB 0,0
11 00050'041000          STA 0,0,2
12 00051'151400          INC 2,2
13 00052'024013-         LDA 1,ITID
14 00053'133000          ADD 1,2
15 00054'050024-         STA 2,.MTRX   ; POINTER - HIT MATRIX
16                  ;  END OF POINTER ASSIGNMENTS
17                  ;*****************************************************************
18 00055'060207 START:NIOC MCAR
19 00056'063507          SKPBZ MCAR
20 00057'000400          JMP .
21 00060'063707          SKPDZ MCAR
22 00061'000400          JMP .
23 00062'060206          NIOC MCAT
24 00063'063506          SKPBZ MCAT
25 00064'000400          JMP .
26 00065'063706          SKPDZ MCAT
27 00066'000400          JMP .
28                  ;  SET UP CONSTANT VALUES IN RECM2 HEADER
29 00067'030264-         LDA 2,.R2HDR
30 00070'020143-         LDA 0,HDCDE
31 00071'041000          STA 0,0,2
32 00072'020006-         LDA 0,TIMCA
33 00073'041001          STA 0,1,2
34 00074'102400          SUB 0,0
35 00075'041003          STA 0,3,2
36                  ;  END OF RECM2 HEADER SECTION
37                  ; SET UP CONSTANT VALUE IN LSVAR TABLE
38 00076'020266-         LDA 0,.R2
39 00077'040000-         STA 0,LSVAP
40                  ; END OF LSVAR SECTION
41 00100'020142-         LDA 0,M16   ; LISTEN FOR RESET HEADER
42 00101'024141-         LDA 1,.RSTHD
43 00102'062007          DOR 0,MCAR
44 00103'065107          DOAS 1,MCAR
45 00104'063607          SKPDN MCAR
46 00105'000777          JMP .-1
47 00106'006152-         JSR @.RCVRV
48 00107'030141-         LDA 2,.RSTHD
49 00110'021000          LDA 0,0,2
50 00111'024143-         LDA 1,HDCDF
51 00112'106404          SUB 0,1,SZR
52 00113'000742          JMP START
53 00114'021001          LDA 0,1,2   ; REMOVE THIS LINE WHEN POLLING LOGIC IS ADDED
54                  ;C      LDA 0,3,2     ; AC0 WILL BE -1 ON RESET
55 00115'101404          INC 0,0,SZR   ; IF RESET HEADER, SKIP
56 00116'000737          JMP START
57 00117'000463          JMP INITL     ; REMOVE THIS LINE WHEN POLLING LOGIC IS ADDED
58 00120'050027-         STA 2,.CURH    ; STORE RESET HEADER ADDRESS INTO CURRENT HEADER PTR
59                  ;*****************************************************************
```

```
  0006  TRKIN
01                ;  RESET LOGIC
02 00121'000461 RESET:JMP INITL   ; THIS WILL BE DELETED WHEN POLLING LOGIC IS ADDED
03                ;RESET:LDA 2,.CURH   ; DELETE ; WHEN POLLING LOGIC IS ADDED
04 00122'021001         LDA 0,1,2
05 00123'040065-        STA 0,RMMCA
06 00124'021002         LDA 0,2,2
07 00125'101004         MOV 0,0,SZR
08 00126'000441         JMP ASSIGN
09 00127'034004-        LDA 3,.READY
10 00130'020006-        LDA 0,TIMCA
11 00131'041401         STA 0,1,3
12 00132'020007-        LDA 0,TITYP
13 00133'041402         STA 0,2,3
14 00134'102400         SUB 0,0
15 00135'041403         STA 0,3,3
16 00136'020270-TR1:    LDA 0,MS
17 00137'024004-        LDA 1,.READY
18 00140'030065-        LDA 2,RMMCA
19 00141'062006         DOR 0,MCAT
20 00142'065006         DOA 1,MCAT
21 00143'073106         DOCS 2,MCAT
22 00144'020142-        LDA 0,M1A
23 00145'024141-        LDA 1,.RSTMD
24 00146'062007         DOR 0,MCAR
25 00147'065107         DOAS 1,MCAR
26 00150'063606         SKRON MCAT
27 00151'000777         JMP .-1
28 00152'030047-        LDA 2,.TR1
29 00153'006151-        JSR @.TRNSV
30 00154'063607         SKRON MCAR
31 00155'000777         JMP .-1
32 00156'006152-        JSR @.RCVSV
33 00157'030141-        LDA 2,.RSTMD
34 00160'021000         LDA 0,0,2
35 00161'024143-        LDA 1,MODDE
36 00162'106404         SUB 0,1,SZR
37 00163'063077         HALT   ; THIS IS NOT A HEADER
38 00164'021003         LDA 0,3,2
39 00165'101404         INC 0,0,SZR
40 00166'063077         HALT   ; THIS IS NOT A RESET HEADER
41 00167'034062-ASSIGN: LDA 3,.CAMAC  ; ASSIGN MCA ADDRESS FROM RESET HEADER
42 00170'021005         LDA 0,5,2
43 00171'041400         STA 0,0,3
44 00172'021006         LDA 0,6,2
45 00173'041401         STA 0,1,3
46 00174'021007         LDA 0,7,2
47 00175'041402         STA 0,2,3
48 00176'021010         LDA 0,10,2
49 00177'041403         STA 0,3,3
50 00200'021013         LDA 0,13,2
51 00201'040066-        STA 0,R2MCA
52                ;  END OF RESET LOGIC
53                ;*************************************************************************
54                ;  INITIALIZE INDICATORS AND POINTERS
55 00202'102400 INITL:SUB 0,0
56 00203'040043-        STA 0,FULL
57 00204'040044-        STA 0,RSTIN
58 00205'040035-        STA 0,.W
59 00206'040042-        STA 0,.HW
```

```
  0007  TRKIN
01 00207'040154-        STA 0,CTRF
02 00210'040157-        STA 0,CTRX
03 00211'040160-        STA 0,CTRY
04 00212'040161-        STA 0,CTRZ
05 00213'020014-        LDA 0,.A
06 00214'040031-        STA 0,.F
07 00215'020015-        LDA 0,.B
08 00216'040032-        STA 0,.X
09 00217'020016-        LDA 0,.C
10 00220'040033-        STA 0,.Y
11 00221'020017-        LDA 0,.D
12 00222'040034-        STA 0,.Z
13 00223'020020-        LDA 0,.HA
14 00224'040036-        STA 0,.HF
15 00225'020021-        LDA 0,.HB
16 00226'040037-        STA 0,.HX
17 00227'020022-        LDA 0,.HC
18 00230'040040-        STA 0,.HY
19 00231'020023-        LDA 0,.HD
20 00232'040041-        STA 0,.HZ
21 00233'020070-        LDA 0,M2
22 00234'040071-        STA 0,CTP
23                ;   FILL 2 DATA & HEADER AREAS
24 00235'006145-FILL2:JSR @.INRTF  ; INPUT TO FILLING BUFFER
25 00236'060177        INTEN       ; INTERRUPT ENABLE
26 00237'020044-WAIT1:LDA 0,RSTIN
27 00240'101004        MOV 0,0,SZR
28 00241'000660        JMP RESET
29 00242'020043-       LDA 0,FULL
30 00243'101005        MOV 0,0,SNR
31 00244'000773        JMP WAIT1
32 00245'006200-       JSR @.SQSMF  ; SQUASH CONTENTS OF AREA F, SET CTRF
33 00246'006144-       JSR @.BUMR  ; BUMR POINTERS, CLEAR FULL
34 00247'010071-       ISZ CTR
35 00250'000765        JMP FILL2
36 00251'006145-       JSR @.INPTF  ; INPUT TO FILLING BUFFER
37 00252'060177        INTEN   ; INTERRUPT ENABLE
38 00253'006146-       JSP @.SETM   ; SET BIT MATRIX FROM X AREA
39 00254'020044-WAIT2:LDA 0,RSTIN
40 00255'101004        MOV 0,0,SZR
41 00256'000643        JMP RESET
42 00257'020043-       LDA 0,FULL
43 00260'101005        MOV 0,0,SNR
44 00261'000773        JMP WAIT2
45                ;***************************************************
46                ;***************************************************
47                ;  FILL NEXT DATA & HEADER AREAS
48 00262'006200-FILLN:JSR @.SQSMF   ; SQUASH CONTENTS OF AREA F, SET CTRF
49 00263'006144-       JSR @.BUMR    ; BUMR POINTERS,CLEAR FULL
50 00264'006145-       JSR @.INPTF   ; INPUT TO FILLING BUFFER
51 00265'060177        INTEN
52 00266'006150-       JSP @.CHCKM   ; COMPARE X & Z AREAS TO BIT MATRIX
53 00267'006146-       JSR @.SETM    ; SET BIT MATRIX FROM X AREA
54 00270'006147-       JSR @.SETHD   ; SET UP HEADER TO PECM2
55 00271'020044-WAITN:LDA 0,RSTIN
56 00272'101004        MOV 0,0,SZR
57 00273'000626        JMP RESET
58 00274'020043-       LDA 0,FULL
59 00275'101005        MOV 0,0,SNR
```

```
  0008  TRKIN
01 00276'000773          JMP WAITN
02 00277'030264-         LDA 2,.R2HDR
03 00300'020072-         LDA 0,TICTR
04 00301'041002          STA 0,2,2  ; STORE TRACK DATA COUNT IN RECM2 HEADER
05 00302'020142-TR2:     LDA 0,M16  ; SEND HEADER TO RECM2
06 00303'024264-         LDA 1,.R2HDR
07 00304'030066-         LDA 2,R2MCA
08 00305'062006          DOB 0,MCAT
09 00306'065006          DOA 1,MCAT
10 00307'073106          DOCS 2,MCAT
11 00310'063606          SKPDN MCAT
12 00311'000777          JMP .-1
13 00312'030073-         LDA 2,.TR2
14 00313'006151-         JSR @.TRNSV
15 00314'020072-TR3:     LDA 0,TICTR
16 00315'030070-         LDA 2,M2  ; ALLOW FOR FIRST 2 DATA CODE WORDS
17 00316'143000          ADD 2,0
18 00317'024266-         LDA 1,.R2
19 00320'030066-         LDA 2,R2MCA    ; SEND TRACK INITIATION DATA TO RECM2
20 00321'062006          DOB 0,MCAT
21 00322'065006          DOA 1,MCAT
22 00323'073106          DOCS 2,MCAT
23 00324'063606          SKPDN MCAT
24 00325'000777          JMP .-1
25 00326'030074-         LDA 2,.TR3
26 00327'006151-         JSR @.TRNSV
27                ;   SEND SQUASHED Y AREA DATA TO RECM2
28 00330'030264-TR4:     LDA 2,.R2HDR
29 00331'021017          LDA 0,17,2
30 00332'024070-         LDA 1,M2
31 00333'123000          ADD 1,0  ; = WORD COUNT -2 OF SQUASHED WORD AREA
32 00334'024033-         LDA 1,.Y
33 00335'030066-         LDA 2,R2MCA
34 00336'062006          DOB 0,MCAT
35 00337'065006          DOA 1,MCAT
36 00340'073106          DOCS 2,MCAT
37 00341'063606          SKPDN MCAT
38 00342'000777          JMP .-1
39 00343'030075-         LDA 2,.TR4
40 00344'006151-         JSR @.TRNSV
41 00345'060477          READS 0
42 00346'024204-         LDA 1,MSKRR
43 00347'107405          AND 0,1,SNR
44 00350'000712          JMP FILLN
45                ; RREPARE LSVAR TAPLE FOR PASSING TO LSXYZ
46 00351'020033-         LDA 0,.Y
47 00352'040001-         STA 0,LSVAR+1
48 00353'020072-         LDA 0,TICTR
49 00354'040002-         STA 0,LSVAR+2
50 00355'020160-         LDA 0,CTRY
51 00356'040003-         STA 0,LSVAR+3
52 00357'006205-         JSR @.LSXYZ  ; PRINT OUT X,Z TRACK INITIATION AND Y AREA DATA
53 00360'000702          JMP FILLN
54                ;  END OF FILLN DATA AND HEADER AREAS LOOP
55                ;*****************************************************************
56                ;*****************************************************************
57                ;  ROUTINE TO INITIALIZE INRUT TO FILLING BUFFER
58 00361'054176-INRTF:STA 3,RTRNF
59 00362'030062-         LDA 2,.CAMAC  ; SET UR WORKING CAM LIST
```

```
  0009  TRAIN
01 00363'030043-       LDA 3,.CAMWK  ; FROM ACCEPTABLE CAM LIST
02 00364'021000        LDA 0,0,2
03 00365'041400        STA 0,0,3
04 00366'021001        LDA 0,1,2
05 00367'041401        STA 0,1,3
06 00370'021002        LDA 0,2,2
07 00371'041402        STA 0,2,3
08 00372'021003        LDA 0,3,2
09 00373'041403        STA 0,3,3
10 00374'102400        SUB 0,0   ; CLEAR FULL INDICATOR
11 00375'040043-       STA 0,FULL
12 00376'020077-       LDA 0,M4   ; SET CAM COUNTER TO -4
13 00377'040101-       STA 0,CMCTR
14 00400'020036-       LDA 0,.HF  ; ESTABLISH CURRENT HEADER POINTER
15 00401'040027-       STA 0,.CURH
16 00402'020031-       LDA 0,.F   ; ESTABLISH CURRENT DATA POINTER
17 00403'040030-       STA 0,.CURD
18 00404'060207        NIOC MCAR  ; UNLOCK RECEIVER
19 00405'020142-       LDA 0,M16     ; LISTEN FOR 1ST HEADER
20 00406'020027-       LDA 1,.CURH
21 00407'062007        DOB 0,MCAR
22 00410'065107        DOAS 1,MCAR
23 00411'000401        JMP .+1
24 00412'002176-       JMP @RTRNF
25                ;  END OF INRTF
26                ;*************************************************************
27                ;*************************************************************
28                ;  ROUTINE TO CLEAR FULL INDICATOR & BUMP POINTERS
29 00413'102400 BUMP: SUB 0,0
30 00414'040043-       STA 0,FULL
31 00415'020034-       LDA 0,.Z
32 00416'040035-       STA 0,.W
33 00417'020033-       LDA 0,.Y
34 00420'040034-       STA 0,.Z
35 00421'020032-       LDA 0,.X
36 00422'040033-       STA 0,.Y
37 00423'020031-       LDA 0,.F
38 00424'040032-       STA 0,.X
39 00425'020035-       LDA 0,.W
40 00426'040031-       STA 0,.F
41 00427'020041-       LDA 0,.HZ
42 00430'040042-       STA 0,.HW
43 00431'020040-       LDA 0,.HY
44 00432'040041-       STA 0,.HZ
45 00433'020037-       LDA 0,.HX
46 00434'040040-       STA 0,.HY
47 00435'020036-       LDA 0,.HF
48 00436'040037-       STA 0,.HX
49 00437'020042-       LDA 0,.HW
50 00440'040036-       STA 0,.HF
51 00441'020160-       LDA 0,CTRY
52 00442'040161-       STA 0,CTRZ
53 00443'020157-       LDA 0,CTRX
54 00444'040160-       STA 0,CTRY
55 00445'020154-       LDA 0,CTRF
56 00446'040157-       STA 0,CTRX
57 00447'102400        SUB 0,0
58 00450'040154-       STA 0,CTRF
59 00451'000401        JMP .+1
```

```
  0010  TRKIN
01 00452'001400         JMP 0,3
02                ;  END OF BUMP
03                ;*****************************************************************************
04                ;*****************************************************************************
05                ;   INTERRUPT SERVICE ROUTINE
06 00453'000401 INTSV:JMP .+1
07 00454'040104-        STA 0,SAVAC  ; SAVE ACCUMULATORS & CARRY BIT
08 00455'044105-        STA 1,SAVAC+1
09 00456'050106-        STA 2,SAVAC+2
10 00457'054107-        STA 3,SAVAC+3
11 00460'101100         MOVL 0,0
12 00461'040110-        STA 0,SAVAC+4
13 00462'020000         LDA 0,0,0
14 00463'040111-        STA 0,SAVIN
15 00464'061477         DIB 0,CPU  ; INTERRUPT ACKNOWLEDGE
16 00465'024112-        LDA 1,SEVEN
17 00466'106404         SUB 0,1,SZR  ; IS INTERRUPT ON MCA RECEIVE?
18 00467'063077         HALT  ; NO
19 00470'060407         DIA 0,MCAR  ; YES
20 00471'040117-        STA 0,RWDAD
21 00472'061407         DIB 0,MCAR
22 00473'040120-        STA 0,RWDCT
23 00474'062407         DIC 0,MCAR
24 00475'040113-        STA 0,STATR
25 00476'024114-        LDA 1,TIMDT
26 00477'107404         AND 0,1,SZR
27 00500'063077         HALT         ; RECEIVER TIME OUT
28 00501'024115-        LDA 1,CNTDR
29 00502'107405         AND 0,1,SNR
30 00503'063077         HALT         ; RECEIVER COUNT NOT DONE
31 00504'030027-        LDA 2,.CURH
32 00505'021000         LDA 0,0,2
33 00506'024103-        LDA 1,HDCDE
34 00507'106404         SUB 0,1,SZR  ; IS THIS A HEADER BLOCK?
35 00510'063077         HALT   ; NO
36 00511'021001         LDA 0,1,2  ; WHEN POLLING LOGIC IS ADDED, THIS WILL BE CHANGED TO LDA 0,3,2
37 00512'101404         INC 0,0,SZP  ; IS THIS A RESET HEADER?
38 00513'000413         JMP CAMCK    ; NO
39 00514'102520         SUBZL 0,0    ; YES
40 00515'040044-        STA 0,RSTIN  ; SET RESET INDICATOR TO 1
41 00516'020110-        LDA 0,SAVAC+4  ; RESTORE REGISTERS
42 00517'101200         MOVR 0,0
43 00520'020104-        LDA 0,SAVAC
44 00521'024105-        LDA 1,SAVAC+1
45 00522'030106-        LDA 2,SAVAC+2
46 00523'034107-        LDA 3,SAVAC+3
47 00524'000401         JMP .+1
48 00525'002111-        JMP @SAVIN   ; RETURN
49 00526'020063-CAMCK:LDA 0,.CAMHK
50 00527'040060-        STA 0,CAMCT
51 00530'020077-        LDA 0,M4
52 00531'040100-        STA 0,CTRI
53 00532'021001         LDA 0,1,2   ; TAKE CAM MCA ADDRESS FROM CURRENT HEADER
54 00533'040271-        STA 0,SVMCA
55 00534'026060-CAMLP:LDA 1,@CAMCT
56 00535'106405         SUB 0,1,SNR
57 00536'000437         JMP CMHIT
58 00537'010060-        ISZ CAMCT
59 00540'010100-        ISZ CTRI
```

```
  0011  TRKIN
01 00541'000773        JMP CAMLR
02 00542'034004-NEGAC:LDA 3,.READY  ;SEND BACK REPLY OF NOT ACCEPTABLE
03 00543'102520        SUBZL 0,0
04 00544'041403        STA 0,3,3
05 00545'020270-TNGCM:LDA 0,MS  ; TRANSMIT ACKNOWLEDGE TO MACHINE WHICH SENT HEADER
06 00546'024004-       LDA 1,.READY
07 00547'030271-       LDA 2,SVMCA
08 00550'062006        DOB 0,MCAT
09 00551'065006        DOA 1,MCAT
10 00552'073106        DOCS 2,MCAT
11 00553'063606        SKPDN MCAT
12 00554'000777        JMP .-1
13 00555'030153-       LDA 2,.INGCM
14 00556'006151-       JSR @.TRNSV
15 00557'060207        NIOC MCAR  ; UNLOCK RECEIVER
16 00560'020142-       LDA 0,M16  ; LISTEN FOR HEADER
17 00561'024027-       LDA 1,.CURM
18 00562'062007        DOB 0,MCAR
19 00563'065107        DOAS 1,MCAR
20 00564'020110-       LDA 0,SAVAC+4  ; RESTORE REGISTERS
21 00565'101200        MOVR 0,0
22 00566'020104-       LDA 0,SAVAC
23 00567'024105-       LDA 1,SAVAC+1
24 00570'030106-       LDA 2,SAVAC+2
25 00571'034107-       LDA 3,SAVAC+3
26 00572'000401        JMP .+1
27 00573'060177        INTEN  ; INTERRUPT ENABLE
28 00574'002111-       JMP @SAVIN   ; RETURN
29 00575'021002 CMMIT:LDA 0,2,2  ; SET UP RECEIVE FOR DATA
30 00576'024070-       LDA 1,M2
31 00577'123000        ADD 1,0
32 00600'024030-       LDA 1,.CURD  ; TO CURRENT DATA POINTER
33 00601'062007        DOB 0,MCAR
34 00602'065107        DOAS 1,MCAR
35 00603'034004-       LDA 3,.READY
36 00604'102400        SUB 0,0
37 00605'041403        STA 0,3,3  ; HEADER IS ACCEPTABLE
38 00606'020270-TROYC: LDA 0,MS  ; TRANSMIT READY MESSAGE TO CAM
39 00607'024004-       LDA 1,.READY
40 00610'030271-       LDA 2,SVMCA
41 00611'062006        DOB 0,MCAI
42 00612'065006        DOA 1,MCAT
43 00613'073106        DOCS 2,MCAT
44 00614'063606        SKPDN MCAT
45 00615'000777        JMP .-1
46 00616'030154-       LDA 2,.TROYC
47 00617'006151-       JSR @.TRNSV
48 00620'063607        SKPON MCAR
49 00621'000777        JMP .-1
50 00622'006152-       JSR @.RCVSV
51 00623'030030-       LDA 2,.CURD
52 00624'020155-       LDA 0,DTCDE
53 00625'025000        LDA 1,0,2
54 00626'122404        SUB 1,0,SZR
55 00627'063077        HALT        ; BLOCK JUST RECEIVED WAS NOT DATA BLOCK
56 00630'102400        SUB 0,0
57 00631'042044-       STA 0,@CAMCT  ; DELETE THIS CAM FROM ACCEPTABLE  LIST
58 00632'010101-       ISZ CMCTR     ; HAVE ALL CAMS SENT DATA?
59 00633'000413        JMP SETUP     ; NO - SET UP RECEIVE FOR MORE
```

```
  0012  TRKIN
01 00634'102520          SUBZL 0,0      ; YES
02 00635'040043-         STA 0,FULL     ; SET FULL INDICATOR TO 1
03 00636'020110-         LDA 0,SAVAC+4  ; RESTORE REGISTERS
04 00637'101200          MOVR 0,0
05 00640'020104-         LDA 0,SAVAC
06 00641'024105-         LDA 1,SAVAC+1
07 00642'030106-         LDA 2,SAVAC+2
08 00643'034107-         LDA 3,SAVAC+3
09 00644'000401          JMP .+1
10 00645'002111-         JMP @SAVIN     ; RETURN
11 00646'020027-SETUP:   LDA 0,.CURH
12 00647'024026-         LDA 1,ICAMH
13 00650'123000          ADD 1,0
14 00651'040027-         STA 0,.CURH
15 00652'020030-         LDA 0,.CURO
16 00653'024025-         LDA 1,ICAMO
17 00654'123000          ADD 1,0
18 00655'040030-         STA 0,.CURO
19 00656'060207          NIOC MCAR      ; UNLOCK MCA RECEIVER
20 00657'020142-         LDA 0,M16      ; SET UP RECEIVE FOR NEXT HEADER
21 00660'024027-         LDA 1,.CURH
22 00661'062007          DOR 0,MCAR
23 00662'065107          DDAS 1,MCAR
24 00663'020110-         LDA 0,SAVAC+4  ; RESTORE REGISTERS
25 00664'101200          MOVR 0,0
26 00665'020104-         LDA 0,SAVAC
27 00666'024105-         LDA 1,SAVAC+1
28 00667'030106-         LDA 2,SAVAC+2
29 00670'034107-         LDA 3,SAVAC+3
30 00671'000401          JMP .+1
31 00672'060177          INTEN          ; INTERRUPT ENABLE
32 00673'002111-         JMP @SAVIN
33                ;*****************************************************************
34                ;*****************************************************************
35                ;  ROUTINE TO SQUASH IN PLACE F AREA DATA FROM 4 CAMS INTO 1 LIST
36 00674'054177-SQSHF:STA 3,RTRNS
37 00675'102400          SUB 0,0
38 00676'040272-         STA 0,NZCAM    ; COUNTER OF NON 0 CAM COUNTS
39 00677'020162-         LDA 0,CTRAD    ; ADDRESS OF LIST OF CAMS' COUNTERS
40 00700'040163-         STA 0,CTRCT
41 00701'020164-         LDA 0,SCMAD    ; ADDRESS OF LIST OF CAMS STARTING ADDRESSES
42 00702'040165-         STA 0,SCMCT
43 00703'020077-         LDA 0,M4
44 00704'040273-         STA 0,ALCAM    ; COUNTER OF ALL CAMS
45 00705'030031-         LDA 2,.F       ; BEGINNING ADDRESS OF F DATA AREA
46 00706'151400          INC 2,2    ; INCREMENT ADDRESS BY 2 TO BYPASS 2
47 00707'151400          INC 2,2     ;  DATA CODE WORDS
48 00710'034034-         LDA 3,.HF   ; BEGINNING ADDRESS OF F HEADER AREA
49 00711'021402 SFLP1:   LDA 0,2,3   ; LOOP TO ESTABLISH 2 LISTS
50 00712'101005          MOV 0,0,SNR    ; 1) STARTING ADDRESSES FOR EACH DATA AREA
51 00713'000411          JMP INCMT
52 00714'024156-         LDA 1,CTRF     ; 2) -WORD COUNT FOR EACH DATA AREA
53 00715'107000          ADD 0,1
54 00716'040156-         STA 1,CTRF
55 00717'042163-         STA 0,@CTRCT   ; CAM(I) NEGATIVE WORD COUNT
56 00720'052165-         STA 2,@SCMCT   ; CAM(I) BEGINNING DATA ADDRESS
57 00721'014272-         DSZ NZCAM
58 00722'010163-         ISZ CTRCT
59 00723'010165-         ISZ SCMCT
```

```
  0013  TRKIN
01 00724'020025-INCMT:LDA 0,ICAMD
02 00725'113000       ADD 0,2
03 00726'020026-      LDA 0,ICAMH
04 00727'117000       ADD 0,3
05 00730'010273-      ISZ ALCAM
06 00731'000760       JMP SFLP1
07 00732'020158-      LDA 1,CTPF
08 00733'125005       MOV 1,1,SNR
09 00734'002177-      JMP @RTPNS    ; RETURN IF THERE IS NO DATA TO BE SQUASHED
10 00735'020162-      LDA 0,CTRAD
11 00736'040163-      STA 0,CTPCT
12 00737'020164-      LDA 0,SCMAD
13 00740'040165-      STA 0,SCMCT
14 00741'126400       SUB 1,1       ; TEMPORARY COUNTER
15 00742'030031-      LDA 2,.F
16 00743'151400       INC 2,2
17 00744'151400       INC 2,2
18 00745'036165-SFLP2:LDA 3,@SCMCT
19 00746'021400 SFLP3:LDA 0,0,3
20 00747'041000       STA 0,0,2
21 00750'175400       INC 3,3
22 00751'151400       INC 2,2
23 00752'125400       INC 1,1    ; TEMPORARY COUNTER
24 00753'012163-      ISZ @CTPCT
25 00754'000772       JMP SFLP3
26 00755'010163-      ISZ CTRCT
27 00756'010165-      ISZ SCMCT
28 00757'010272-      ISZ NZCAM
29 00760'000765       JMP SFLP2
30 00761'020156-      LDA 0,CTRF
31 00762'123004       ADD 1,0,SZR
32 00763'063077       HALT        ; # OF DATA WORDS SENSED IN F
33 00764'000401       JMP .+1     ; DOESN'T AGREE WITH SUM OF HEADER COUNTS
34 00765'002177-      JMP @RTPNS
35                ;  END OF SQUASH ROUTINE
36                ;*************************************************************************
37                ;*************************************************************************
38                ;  ROUTINE TO SET UP HEADER FOR PECM2
39 00766'054201-SETHD:STA 3,RTPNH
40 00767'030264-      LDA 2,.P2HDR
41 00770'102400       SUB 0,0
42 00771'041002       STA 0,2,2
43 00772'041012       STA 0,12,2
44 00773'030037-      LDA 3,.HX   ; STORE X AREA VARIABLES
45 00774'021402       LDA 0,2,3   ; # OF LEAKERS FROM EACH CAM IN X AREA
46 00775'041004       STA 0,4,2
47 00776'021422       LDA 0,22,3
48 00777'041006       STA 0,6,2
49 01000'021442       LDA 0,42,3
50 01001'041010       STA 0,10,2
51 01002'021462       LDA 0,62,3
52 01003'041012       STA 0,12,2
53 01004'021401       LDA 0,1,3   ; MCA ADDRESSES OF EACH CAM IN X AREA
54 01005'105220       MOVZR 0,1
55 01006'125220       MOVZR 1,1
56 01007'125220       MOVZR 1,1
57 01010'125220       MOVZR 1,1
58 01011'021421       LDA 0,21,3
59 01012'100000       COM 0,0    ; LOGICAL OR
```

```
  0014  TRKIN
01 01013'107400          AND 0,1
02 01014'106220          ADCZR 0,1
03 01015'125220          MOVZR 1,1
04 01016'125220          MOVZR 1,1
05 01017'125220          MOVZR 1,1
06 01020'021441          LDA 0,41,3
07 01021'100000          COM 0,0
08 01022'107400          AND 0,1
09 01023'106220          ADCZR 0,1
10 01024'125220          MOVZR 1,1
11 01025'125220          MOVZR 1,1
12 01026'125220          MOVZR 1,1
13 01027'021461          LDA 0,61,3
14 01030'100000          COM 0,0
15 01031'107400          AND 0,1
16 01032'106000          ADC 0,1
17 01033'045014          STA 1,14,2     ; COMPOSITE WORD OF MCA ADDRESSES
18 01034'021405          LDA 0,5,3      ; FROM X AREA
19 01035'041015          STA 0,15,2     ; FRAME # OF X AREA DATA
20 01036'034041-         LDA 3,.MZ      ; STORE Z AREA VARIABLE
21 01037'021402          LDA 0,2,3      ; # OF LEAKERS FROM EACH CAM IN Z AREA
22 01040'041005          STA 0,5,2
23 01041'021422          LDA 0,22,3
24 01042'041007          STA 0,7,2
25 01043'021442          LDA 0,42,3
26 01044'041011          STA 0,11,2
27 01045'021462          LDA 0,62,3
28 01046'041013          STA 0,13,2
29 01047'021405          LDA 0,5,3    ; FRAME # OF Z AREA DATA
30 01050'041016          STA 0,16,2
31 01051'020160-         LDA 0,CTRY    ; -WORD COUNT OF DATA IN Y AREA
32 01052'041017          STA 0,17,2
33 01053'000401          JMP .+1
34 01054'002201-         JMP @RTRNM
35                ;  END OF ROUTINE TO SET UP HEADER FOR RECM2
36                ;*****************************************************************
37                ;*****************************************************************
38                ;  ROUTINE TO SET UP BIT MATRIX FROM DATA IN X AREA
39 01055'054202-SETM: STA 3,RTRNM
40 01056'020157-         LDA 0,CTRX  ; TEST CTRX FOR NON-ZERO VALUE
41 01057'101005          MOV 0,0,SNR
42 01060'001400          JMP 0,3     ; IF CTRX=0, RETURN
43 01061'034227-         LDA 3,M12A  ; CLEAR BIT MATRIX OF 256 BY 16 WORDS
44 01062'030024-         LDA 2,.MTRX
45 01063'024224-         LDA 1,B32
46 01064'102400          SUB 0,0
47 01065'041000 CLRLP:STA 0,0,2
48 01066'041001          STA 0,1,2
49 01067'041002          STA 0,2,2
50 01070'041003          STA 0,3,2
51 01071'041004          STA 0,4,2
52 01072'041005          STA 0,5,2
53 01073'041006          STA 0,6,2
54 01074'041007          STA 0,7,2
55 01075'041010          STA 0,10,2
56 01076'041011          STA 0,11,2
57 01077'041012          STA 0,12,2
58 01100'041013          STA 0,13,2
59 01101'041014          STA 0,14,2
```

```
  0015  TRKIN
01 01102'041015        STA 0,15,2
02 01103'041016        STA 0,16,2
03 01104'041017        STA 0,17,2
04 01105'041020        STA 0,20,2
05 01106'041021        STA 0,21,2
06 01107'041022        STA 0,22,2
07 01110'041023        STA 0,23,2
08 01111'041024        STA 0,24,2
09 01112'041025        STA 0,25,2
10 01113'041026        STA 0,26,2
11 01114'041027        STA 0,27,2
12 01115'041030        STA 0,30,2
13 01116'041031        STA 0,31,2
14 01117'041032        STA 0,32,2
15 01120'041033        STA 0,33,2
16 01121'041034        STA 0,34,2
17 01122'041035        STA 0,35,2
18 01123'041036        STA 0,36,2
19 01124'041037        STA 0,37,2
20 01125'133000        ADD 1,2
21 01126'175404        INC 3,3,SZR
22 01127'000736        JMP CLRLP
23 01130'000401        JMP .+1
24 01131'034032-       LDA 3,.X    ; AC3 CONTAINS BASE ADDRESS OF x  DATA AREA
25 01132'175400        INC 3,3
26 01133'175400        INC 3,3
27 01134'020157-       LDA 0,CTRX
28 01135'040230-       STA 0,CTRM
29 01136'030024-PATLP: LDA 2,.MTRX  ; AC2 CONTAINS BASE ADDRESS OF BIT MATRIX
30 01137'021401        LDA 0,1,3   ; TAKE Y VALUE
31 01140'101120        MOVZL 0,0     ; SHIFT LEFT 2 BITS
32 01141'101120        MOVZL 0,0
33 01142'024231-       LDA 1,MSKL4   ; MASK OFF LOW 4 BITS
34 01143'123400        AND 1,0
35 01144'113000        ADC 0,2       ; ADD TO BIT MATRIX BASE ADDRESS
36 01145'000401        JMP .+1
37                                   ; AC2 CONTAINS NEW Y ADDRESS
38 01146'021400        LDA 0,0,3     ; TAKE x VALUE
39 01147'101220        MOVZR 0,0   ; SHIFT RIGHT 2 BITS
40 01150'101220        MOVZR 0,0
41 01151'024233-       LDA 1,MKU12
42 01152'107400        AND 0,1       ; MASK OFF UPPER 12 BITS
43 01153'054235-       STA 3,SVXAD
44 01154'034232-       LDA 3,.PATRN
45 01155'137000        ADD 1,3       ; ADD TO BIT PATTERN BASE ADDRESS
46 01156'024234-       LDA 1,MKR11   ; MASK OFF BITS 8-11
47 01157'107620        ANDZR 0,1     ; SHIFT RIGHT 4 BITS
48 01160'125220        MOVZR 1,1
49 01161'125220        MOVZR 1,1
50 01162'125220        MOVZR 1,1
51 01163'133000        ADD 1,2     ; ADD X LOCATION OFFSET TO Y LOCATION
52 01164'021400        LDA 0,0,3
53 01165'025000        LDA 1,0,2
54 01166'100000        COM 0,0     ; LOGICAL OR
55 01167'107400        AND 0,1
56 01170'106000        ADC 0,1
57 01171'045000        STA 1,0,2   ; IOR BIT PATTERN IN BIT MATRIX
58 01172'000401        JMP .+1
59 01173'034235-       LDA 3,SVXAD
```

```
  0016  TRKIN
01 01174'175400         INC 3,3
02 01175'175400         INC 3,3
03 01176'010230-        ISZ CTRM
04 01177'000002         JMP .+2
05 01200'063077         HALT    ; ODD COUNTER FOR X AREA DATA
06 01201'010230-        ISZ CTRM
07 01202'000734         JMP PATLR
08 01203'000401         JMP .+1
09 01204'002202-        JMP @RTRNM
10                ;  END OF SETM ROUTINE
11                ;*************************************************************
12                ;*************************************************************
13                ;  ROUTINE TO CHECK DATA POINTS IN AREAS X & Z
14                ;     FOR A MATCH IN THE BIT MATRIX, THEREBY
15                ;     INDICATING A TRACK IS DETECTED.
16 01205'054203-CHCKM:STA 3,RTRNC
17 01206'102400         SUB 0,0
18 01207'040072-        STA 0,TICTR
19 01210'020157-        LDA 0,CTRX    ; TEST CTRX, CTRY, CTRZ
20 01211'101005         MOV 0,0,SNR   ; IF ANY ONE IS 0, RETURN
21 01212'001400         JMP 0,3
22 01213'020160-        LDA 0,CTRY
23 01214'101005         MOV 0,0,SNR
24 01215'001400         JMP 0,3
25 01216'020161-        LDA 0,CTRZ
26 01217'101005         MOV 0,0,SNR
27 01220'001400         JMP 0,3
28 01221'020266-        LDA 0,.R2
29 01222'101400         INC 0,0
30 01223'101400         INC 0,0
31 01224'040245-        STA 0,R2CT
32 01225'020032-        LDA 0,.X
33 01226'101400         INC 0,0
34 01227'101400         INC 0,0
35 01230'040241-        STA 0,XCT
36 01231'020157-        LDA 0,CTRX
37 01232'040236-        STA 0,MCTR1
38 01233'020161-SRCHX:LDA 0,CTRZ
39 01234'040237-        STA 0,MCTR2
40 01235'020034-        LDA 0,.Z
41 01236'101400         INC 0,0
42 01237'101400         INC 0,0
43 01240'040242-        STA 0,ZCT
44 01241'030241-SRCHZ:LDA 2,XCT
45 01242'034242-        LDA 3,ZCT
46 01243'021001         LDA 0,1,2     ; Y VALUE IN X AREA
47 01244'025401         LDA 1,1,3     ; Y VALUE IN Z AREA
48 01245'107120         ADDZL 0,1
49 01246'020231-        LDA 0,MSKL4
50 01247'107403         AND 0,1       ; MASK OFF LOW 4 BITS
51 01250'020024-        LDA 0,.MTRX
52 01251'123000         ADD 1,0       ; ADD TO BASE ADDRESS OF BIT MATRIX
53 01252'040240-        STA 0,MTXCT
54 01253'021000         LDA 0,0,2     ; X VALUE IN X AREA
55 01254'025400         LDA 1,0,3     ; X VALUE IN Z AREA
56 01255'107220         ADDZR 0,1
57 01256'125220         MOVZR 1,1
58 01257'125220         MOVZR 1,1
59 01260'020233-        LDA 0,MKU12   ; MASK OFF UPPER 12 BITS
```

```
  0017  TRKIN
01 01261'123400        AND 1,0
02 01262'034232-       LDA 3,.RATRN
03 01263'117000        ADD 0,3          ; ADDRESS OF BIT RATTERN WORD
04 01264'020234-       LDA 0,MKB11
05 01265'123620        ANDZR 1,0
06 01266'101220        MOVZR 0,0
07 01267'101220        MOVZR 0,0
08 01270'101220        MOVZR 0,0
09 01271'030240-       LDA 2,MTXCT
10 01272'113000        ADD 0,2          ; ADDRESS IN BIT MATRIX
11 01273'021000        LDA 0,0,2        ; ACTUAL VALUE IN BIT MATRIX
12 01274'025400        LDA 1,0,3      ; BIT RATTERN WORD
13 01275'107405        AND 0,1,SNR
14 01276'000431        JMP INCRZ        ; NO MATCH FOUND
15 01277'000401        JMR .+1          ; MATCH FOUND
16 01300'030241-       LDA 2,XCT        ; STORE X & Y COORDINATE PAIR
17 01301'034265-       LDA 3,R2CT       ; FROM X AREA & Z AREA IN
18 01302'021000        LDA 0,0,2        ; TRACK INITTATION DATA LIST
19 01303'041400        STA 0,0,3
20 01304'021001        LDA 0,1,2
21 01305'041401        STA 0,1,3
22 01306'030242-       LDA 2,ZCT
23 01307'021000        LDA 0,0,2
24 01310'041402        STA 0,2,3
25 01311'021001        LDA 0,1,2
26 01312'041403        STA 0,3,3
27 01313'024267-       LDA 1,P4       ; INCREMENT TRACK INITIATION DATA ADDRESS
28 01314'137000        ADD 1,3
29 01315'054265-       STA 3,R2CT
30 01316'020072-       LDA 0,TICTR
31 01317'024077-       LDA 1,M4       ; ADD -4 TO TRACK INITTATION COUNTER
32 01320'107000        ADD 0,1
33 01321'044072-       STA 1,TICTR
34 01322'020027        LDA 0,M1024
35 01323'122405        SUB 1,0,SNR
36 01324'002203-       JMP @RTRNC
37 01325'101123        MOVZL 0,0,SNC
38 01326'063077        HALT             ; TICTR OVERFLOW
39 01327'010242-INCRZ:ISZ ZCT
40 01330'010242-       ISZ ZCT
41 01331'000401        JMP .+1
42 01332'010237-       ISZ MCTR2
43 01333'000402        JMP .+2
44 01334'063077        HALT    ; ODD COUNTER FOR Z AREA DATA
45 01335'010237-       ISZ MCTR2
46 01336'000703        JMR SRCHZ
47 01337'000401        JMR .+1
48 01340'010241-       ISZ XCT
49 01341'010241-       ISZ XCT
50 01342'010236-       ISZ MCTR1
51 01343'000402        JMR .+2
52 01344'063077        HALT    ; ODD COUNTER FOR X AREA DATA
53 01345'010236-       ISZ MCTR1
54 01346'000665        JMP SRCHX
55 01347'000401        JMP .+1
56 01350'002203-       JMR @RTRNC
57 01351'176000 M1024:-1024.
58              ;  END OF CHCKM ROUTINE
59              ;*************************************************************
```

```
  001H  TRKIN
01              ;*************************************************************
02       000000'      .END TRKIN    ;  END OF TRACK INITIATION PROGRAM
```

```
0019  TRKIN

ALCAM   000273-     4/26    12/44   13/05
ASSIG   000167'     6/08    6/41
BUMP    000413'     2/54    9/29
CAMAC   000052-     2/11    2/19
CAMCK   000526'    10/38   10/49
CAMCT   000064-     2/21    10/50   10/55   10/58   11/57
CAMLP   000534'    10/55   11/01
CAMWK   00005A-     2/15    2/20
CHCKM   001205'     2/58    16/16
CLRLP   001065'    14/47   15/22
CMCTR   000101-     2/34    9/13    11/58
CMHIT   000575'    10/57   11/29
CNTDR   000116-     2/47    10/28
CNTDT   000115-     2/46
CTR     000071-     2/26    7/22    7/34
CTRAD   000162-     3/09    12/39   13/10
CTRCT   000163-     3/10    12/40   12/55   12/58   13/11   13/24   13/26
CTRF    000156-     3/05    7/01    9/55    9/58    12/52   12/54   13/07   13/30
CTRI    000100-     2/33    10/52   10/59
CTRLS   000166-     3/09    3/13
CTRM    000230-     3/41    15/28   16/03   16/06
CTRX    000157-     3/06    7/02    9/53    9/56    14/40   15/27   16/19   16/36
CTRY    000160-     3/07    7/03    8/50    9/51    9/54    14/31   16/22
CTRZ    000161-     3/08    7/04    9/52    16/25   16/38
DTCDE   000155-     3/04    5/07    11/52
FILL2   000235'     7/24    7/35
FILLN   000262'     7/48    8/44    8/53
FULL    000043-     2/01    6/56    7/29    7/42    7/58    9/11    9/30    12/02
HDCDE   000143-     2/53    5/30    5/50    6/35    10/33
ICAMD   000025-     1/46    12/16   13/01
ICAMH   000026-     1/47    12/12   13/03
IDATA   000011-     1/32    4/48
IHDR    000012-     1/34    4/57
INCMT   000724'    12/51   13/01
INCRZ   001327'    17/14   17/39
INITL   000202'     5/57    6/02    6/55
INPIF   000361'     2/55    8/58
INTAD   000102-     2/35    4/30
INTMK   000103-     2/36    4/33
INTSV   000453'     2/35    10/06
ITID    000013-     1/35    5/13
LSVAR   000000-     1/21    5/39    8/47    8/49    8/51
LSXYZ   000205-X    3/22
M1024   001351'    17/34   17/57
M12A    000227-     3/40    14/43
M16     000142-     2/52    5/41    6/22    8/05    9/19    11/16   12/20
M2      000070-     2/25    7/21    8/16    8/30    11/30
M4      000077-     2/32    9/12    10/51   12/43   17/31
M5      000270-     4/23    6/16    11/05   11/38
MCAMK   000005-     1/28    4/39
MCTR1   000236-     3/47    14/37   17/50   17/53
MCTR2   000237-     3/48    16/39   17/42   17/45
MKA11   000234-     3/45    15/46   17/04
MKU12   000233-     3/44    15/41   16/59
MSKL4   000231-     3/42    15/33   16/49
MSKPR   000204-     3/21    8/42
MTXCT   000240-     3/49    16/53   17/09
NFGAC   000542'    11/02
MZCAM   000272-     4/25    12/38   12/57   13/28
```

```
0020  TRKIN

P32       000226-      3/39    14/45
PU        000267-      4/22    17/27
PATLP     001136'     15/29    16/07
PATRN     00020A-      3/23     3/43
PMMCA     000065-      2/22     6/05    6/18
P2CT      000245-      4/20    16/31   17/17   17/29
P2HDQ     000248-      4/18     4/19
P2MCA     000064-      2/23     6/51    8/07    8/19    8/33
PCVSV     000152-X     3/01
READY     000045-      1/27     2/0A
RESET     000121'      6/02     7/2A    7/41    7/57
RSTHD     000121-      2/50     2/51
RSTTM     000048-      2/04     6/57    7/26    7/39    7/55   10/40
RTRN      000074-      2/31
RTRNC     000203-      3/20    16/14   17/36   17/5A
RTRNF     00017A-      3/15     8/5A    9/2A
RTRNH     000201-      3/18    13/30   14/34
RTRNM     000202-      3/19    14/39   16/09
RTRNR     000177-      3/16    12/36   13/09   13/34
RWDAD     000117-      2/48    10/20
RWDCT     000120-      2/49    10/22
SAVAC     000104-      2/37    10/07   10/0A   10/09   10/10   10/12   10/41   10/43
                      10/44   10/45   10/4A   11/20   11/22   11/23   11/24   11/25
                      12/03   12/05   12/0A   12/07   12/0A   12/24   12/26   12/27
                      13/2A   12/29
SAVIN     000111-      2/42    10/14   10/4A   11/2A   12/10   12/32
SCMAD     000164-      3/11    12/41   13/12
SCMCT     000165-      3/12    12/42   12/56   12/59   13/13   13/1A   13/27
SCMLS     000172-      3/11     3/14
SETHD     000746'      2/57    13/39
SETIM     031055'      2/56    14/39
SETUP     000646'     11/59    12/11
SEVEN     000112-      2/43    10/1A
SFLP1     000711'     12/49    13/0A
SFLP2     000745'     13/1A    13/2B
SFLP3     000746'     13/19    13/25
SDSHF     000674'      3/17    12/3A
SRCHX     001233'     16/3A    17/5A
SWCHZ     001241'     16/4A    17/4A
START     000055'      5/1A     5/52    5/56
STATH     000113-      2/44    10/2A
SYMCA     000271-      4/24    10/5A   11/07   11/40
SYXAD     000235-      3/4A    15/43   15/59
TICTR     000072-      2/27     6/03    8/15    8/4A   16/1A   17/30   17/33
TIMCA     000006-      1/23     4/41    5/32    6/10
TIMCT     000114-      2/45    10/25
TITYP     000007-      1/30     6/12
TKGEN     000545'      3/02    11/05
TP1       000136'      2/24     6/1A
TP2       000302'      2/28     8/05
TP3       000310'      2/29     8/15
TP4       000330'      2/30     8/2A
TPDYC     000606'      3/03    11/3A
TPKIN     000000'      4/2A    18/02
TPNSV     000151-X     2/59
WAIT1     000237'      7/26     7/31
WAIT2     000254'      7/39     7/44
WAITN     000271'      7/55     8/01
XCT       000241-      3/50    16/35   16/44   17/16   17/4B   17/49
```

```
0021  TRKIN

YCT     000243-     3/52
ZCT     000242-     3/51    16/43   16/45   17/22   17/39   17/40
.A      000010-     1/37    4/47    7/05
.B      000015-     1/38    4/50    7/07
.BUMP   000144-     2/54    7/33    7/49
.C      000016-     1/39    4/52    7/09
.CAMA   000042-     2/19    6/41    8/59
.CAMW   000043-     2/20    9/01    10/49
.CHCK   000150-     2/58    7/52
.CURD   000030-     1/49    9/17    11/32   11/51   12/15   12/18
.CURM   000027-     1/48    5/58    9/15    9/20    10/31   11/17   12/11   12/14
                    12/21
.D      000017-     1/40    4/54    7/11
.E      000031-     1/50    7/06    9/16    9/37    9/40    12/45   13/15
.HA     000020-     1/41    4/56    7/13
.HH     000021-     1/42    4/59    7/15
.HC     000022-     1/43    5/02    7/17
.HD     000023-     1/44    5/04    7/19
.HE     000036-     1/55    7/14    9/14    9/47    9/50    12/48
.HW     000042-     1/59    6/59    9/42    9/49
.HX     000037-     1/56    7/16    9/45    9/48    13/44
.HY     000040-     1/57    7/18    9/43    9/46
.HZ     000041-     1/58    7/20    9/41    9/44    14/20
.INPT   000145-     2/55    7/24    7/36    7/50
.LSXY   000205-     3/22    8/52
.MTPX   000020-     1/45    5/15    14/44   15/29   16/51
.NMAX   000010-     1/31    4/45
.PATN   000232-     3/43    15/44   17/02
.R2     000266-     4/21    5/06    5/38    8/18    16/28
.R2MD   000244-     4/19    5/29    8/02    8/06    8/28    13/40
.RCYS   000152-     3/01    5/47    6/32    11/50
.READ   000004-     1/27    4/36    6/09    6/17    11/02   11/06   11/35   11/39
.RSTH   000141-     2/51    5/42    5/48    6/23    6/33
.SETH   000147-     2/57    7/54
.SETW   000146-     2/56    7/38    7/53
.SOSW   000200-     3/17    7/32    7/48
.TAGC   000153-     3/02    11/13
.TR1    000067-     2/28    6/28
.TR2    000073-     2/28    8/13
.TR3    000074-     2/29    8/25
.TR4    000075-     2/30    8/39
.TRDY   000154-     3/03    11/46
.TPNS   000151-     2/59    6/20    8/14    8/26    8/40    11/14   11/47
.X      000035-     1/54    6/58    9/32    9/39
.Y      000032-     1/51    7/08    9/35    9/38    15/24   16/32
.Y      000033-     1/52    7/10    8/32    8/46    9/33    9/36
.Z      000034-     1/53    7/12    9/31    9/34    16/40
```

```
  0001  LSXYZ                LSXYZ2
01                      .TITL LSXYZ
02                      .ENT LSXYZ
03          000001      .IXTM 1
04                  ;*****************************************************************
05                  ;*****************************************************************
06                  ;   SUBROUTINE TO OUTPUT X AND Z TRACK INITIATION DATA,
07                  ;      AND ALL OF Y AREA DATA TO THE TELETYPE.
08                  ;      ZERO PAGE LOCATIONS 50 - 53 ARE USED AS
09                  ;      THE LINK FOR 4 POINTERS.  THESE LOCATIONS WILL BE SET BY THE
10                  ;      CALLING PROGRAM, TRKIN.
11                  ;        Z.P. 50 = POINTER TO T.I. AREA
12                  ;        Z.P. 51 = POINTER TO Y AREA
13                  ;        Z.P. 52 = - WORD COUNT OF T.I. DATA
14                  ;        Z.P. 53 = - WORD COUNT OF Y DATA
15                  ;*****************************************************************
16                  ;*****************************************************************
17                      .ZREL
18 00000-000000 R2CT:0
19 00001-000000 RTRNO:0
20 00002-000000 XZCTR:0
21 00003-000000 YCTR:0
22 00004-000000 LN4D:0
23 00005-000000 YCT:0
24                      .NREL
25 00000'177747 M9:-9.
26 00001'000000 .R2:0     ; POINTER - TRACK INITIATION DATA
27 00002'000000 .Y:0      ; POINTER - Y AREA DATA
28 00003'000000 TICTR:0 ; -WORD COUNT - T.I.DATA
29 00004'000000 CTRY:0   ; -WORD COUNT - Y  DATA
30 00005'000417'LN1AD:LINE1
31 00006'000455'LN2AD:LINE2
32 00007'000607'LN5AD:LINE5
33 00010'000275'.LNOT:LNOUT
34 00011'054001-LSXYZ:STA 3,RTRNO
35 00012'020050       LDA 0,50.0
36 00013'040766       STA 0,.R2
37 00014'020051       LDA 0,51.0
38 00015'040765       STA 0,.Y
39 00016'020052       LDA 0,52.0
40 00017'040764       STA 0,TICTR
41 00020'020053       LDA 0,53.0
42 00021'040763       STA 0,CTRY
43 00022'060211       NIOC TTO    ; CLEAR TELETYPE OUTPUT
44 00023'020760       LDA 0,TICTR   ; TEST TICTR & CTRY
45 00024'101004       MOV 0,0,SZR   ; IF BOTH ARE ZERO, PRINT MESSAGE & RETURN
46 00025'000410       JMP HAVDT
47 00026'020756       LDA 0,CTRY
48 00027'101004       MOV 0,0,SZR
49 00030'000405       JMP HAVDT
50 00031'020756       LDA 0,LN5AD
51 00032'024746       LDA 1,M9
52 00033'006755       JSR @.LNOT
53 00034'002001-      JMP @RTRNO
54 00035'020744 HAVDT:LDA 0,.R2
55 00036'101400       INC 0,0
56 00037'101400       INC 0,0
57 00040'040000-       STA 0,R2CT
58 00041'020741       LDA 0,.Y
59 00042'101400       INC 0,0
```

```
  0002  LSXYZ
01 00003'101400         INC 0,0
02 00004'040005-        STA 0,YCT
03 00005'102400         SUB 0,0
04 00006'040512         STA 0,ENDXZ  ; IF ENDXZ=0 THERE IS MORE X,Z TRACK INTITIATION
05 00007'040513         STA 0,ENDY   ;                DATA TO BE PRINTED.
06                                   ; IF ENDXZ=1, THERE IS NO MORE X,Z TI DATA.
07                                   ; ENDY ARRLIES THE SAME WAY TO Y AREA DATA.
08 00050'020733         LDA 1,TICTR
09 00051'125004         MOV 1,1,SZR
10 00052'000403         JMP .+3
11 00053'102520         SUBZL 0,0
12 00054'040504         STA 0,ENDXZ
13 00055'040002-        STA 1,XZCTR
14 00056'020726         LDA 1,CTRY
15 00057'125004         MOV 1,1,SZR
16 00060'000403         JMP .+3
17 00061'102520         SUBZL 0,0
18 00062'040477         STA 0,ENDY
19 00063'040003-        STA 1,YCTR
20 00064'020721         LDA 0,LN1AD  ; PRINT OUT THE HEADER LINE
21 00065'020475         LDA 1,M2R
22 00066'006722         JSR @.LNOT
23 00067'030717 LN2AR:  LDA 3,LN2AD  ; PRINT LISTS OF THE X TI VARIABLES,
24 00070'050000-        STA 3,LNAD
25 00071'020467         LDA 0,ENDXZ  ;  THE Z TI VARIABLES, AND THE Y AREA
26 00072'101000         MOV 0,0,SZR  ;  DATA VARIABLES.
27 00073'000405         JMP CHKY
28                                       ; IF ENDXZ=0 & ENDY=0, STAY IN LN2AR
29                                       ; IF ENDXZ=1 & ENDY=0, GO TO LN3AR
30                                       ; IF ENDXZ=0 & ENDY=1, GO TO LN4AR
31                                       ; IF ENDXZ=1 & ENDY=1, RETURN.  ALL DONE.
32 00074'020465         LDA 0,ENDY
33 00075'101000         MOV 0,0,SZR
34 00076'000525         JMP LN4AR
35 00077'000405         JMP .+5
36 00100'020461 CHKY:   LDA 0,ENDY
37 00101'101000         MOV 0,0,SZR
38 00102'002001-        JMP @RTRND
39 00103'000461         JMP LN3AR
40 00104'020457         LDA 0,M4
41 00105'040556         STA 0,PCTR
42 00106'022000-RLOP2:  LDA 0,@R2CT
43 00107'004555         JSR @.PCONV      ; CONVERT WORD TO ASCII CHARACTERS
44 00110'030000-        LDA 3,LNAD
45 00111'175400         INC 3,3
46 00112'175400         INC 3,3
47 00113'050000-        STA 3,LNAD
48 00114'010000-        ISZ R2CT
49 00115'010546         ISZ PCTR
50 00116'000770         JMP RLOP2
51 00117'020002-        LDA 0,XZCTR
52 00120'024545         LDA 1,M4
53 00121'107000         ADD 0,1,SZR
54 00122'000405         JMP NEGT1
55 00123'040002-        STA 1,XZCTR
56 00124'126520         SUBZL 1,1
57 00125'040433         STA 1,ENDXZ
58 00126'000400         JMP YOUT
59 00127'040002-NEGT1:  STA 1,XZCTR
```

```
  0003  LSXYZ
01 00130'125133         MOVZL# 1,1,SNC
02 00131'063077         HALT                    ; IF COUNT IS NOT ZERO, IT MUST BE NEGATIVE
03 00132'022005-YOUT:   LDA 0,@YCT
04 00133'006531         JSR @.RCONV
05 00134'034004-        LDA 3,LNAD
06 00135'175400         INC 3,3
07 00136'175400         INC 3,3
08 00137'054004-        STA 3,LNAD
09 00140'010005-        ISZ YCT
10 00141'022005-        LDA 0,@YCT
11 00142'006522         JSR @.RCONV
12 00143'010005-        ISZ YCT
13 00144'010003-        ISZ YCTR
14 00145'000402         JMP .+2
15 00146'063077         HALT            ; THIS INDICATES AN 000 Y COUNT, WHICH IS WRONG
16 00147'010003-        ISZ YCTR
17 00150'000403         JMP .+3
18 00151'126520         SUBZL 1,1
19 00152'044007         STA 1,ENDY
20 00153'000401         JMP .+1
21 00154'020032         LDA 0,LN2AD     ; OUTPUT CONTENTS OF LINE 2
22 00155'024005         LDA 1,M29
23 00156'004517         JSR LNOUT
24 00157'000710         JMP LN2AR
25 00160'000000 ENDXZ:0
26 00161'000000 ENDY:0
27 00162'177743 M29:-29.
28 00163'177774 M4:-4.
29 00164'034003 LN3AR:LDA 3,LN3AD       ; LOOP HERE WHEN X,Z II DATA HAS BEEN EXHAUSED,
30 00165'054004-        STA 3,LNAD
31 00166'020773         LDA 0,ENDY      ;  BUT Y DATA STILL EXISTS
32 00167'101004         MOV 0,0,SZR
33 00170'002001-        JMP @RTRNO
34 00171'020775         LDA 0,YOFST     ; YOFST=# OF WORDS TO SKIP TO REACH Y LOCATION
35 00172'177000         ADD 0,3
36 00173'054004-        STA 3,LNAD
37 00174'022005-        LDA 0,@YCT
38 00175'006467         JSR @.RCONV
39 00176'034004-        LDA 3,LNAD
40 00177'175400         INC 3,3
41 00200'175400         INC 3,3
42 00201'054004-        STA 3,LNAD
43 00202'010005-        ISZ YCT
44 00203'022005-        LDA 0,@YCT
45 00204'006530         JSR RCONV
46 00205'010005-        ISZ YCT
47 00206'010003-        ISZ YCTR
48 00207'000402         JMP .+2
49 00210'063077         HALT            ; THIS INDICATES AN 000 Y COUNT, WHICH IS WRONG
50 00211'010003-        ISZ YCTR
51 00212'000403         JMP .+3
52 00213'126520         SUBZL 1,1
53 00214'044745         STA 1,ENDY
54 00215'000401         JMP .+1
55 00216'020451         LDA 0,LN3AD
56 00217'024743         LDA 1,M29
57 00220'004455         JSR LNOUT
58 00221'000401         JMP .+1
59 00222'000742         JMP LN3AR
```

```
  0000  LSXYZ
01 00223'030005 LN0AR:LDA 3,LN0AD  ; LOOP HERE WHEN Y DATA HAS BEEN EXHAUSTED,
02 00224'054004-       STA 3,LNAD
03 00225'020733        LDA 0,ENDXZ  ;  OUT X,Z TI DATA STILL EXISTS.
04 00226'101004        MOV 0,0,SZR
05 00227'002001-       JMP #NIRNO
06 00230'020733        LDA 0,M0
07 00231'040032        STA 0,PCTR
08 00232'022000-PLOP0:LDA 0,#R2CT
09 00233'004031        JSR @.PCONV
10 00234'034004-       LDA 3,LNAD
11 00235'175000        INC 3,3
12 00236'175000        INC 3,3
13 00237'054004-       STA 3,LNAD
14 00240'010000-       ISZ R2CT
15 00241'010032        ISZ PCTR
16 00242'000770        JMP PLOP0
17 00243'020002-       LDA 0,XZCTR
18 00244'024021        LDA 1,P0
19 00245'107000        ADD 0,1,SZR
20 00246'000005        JMP NEGT2
21 00247'040002-       STA 1,XZCTR
22 00250'126520        SUBZL 1,1
23 00251'040707        STA 1,ENDXZ
24 00252'000000        JMP .+0
25 00253'040002-NEGT2:STA 1,XZCTR
26 00254'125133        MOVZL# 1,1,SNC
27 00255'063077        HALT            ; IF COUNT IS NOT ZERO, IT MUST BE NEGATIVE.
28 00256'020012        LDA 0,LN0AD
29 00257'024703        LDA 1,M29
30 00260'000015        JSR LNOUT
31 00261'000001        JMP .+1
32 00262'000741        JMP LN0AR
33                ;  END OF ROUTINE TO OUTPUT X,Z TI DATA & Y AREA DATA
34                ;*************************************************************************
35 00263'000000 PCTR:0
36 00264'000330'.PCONV:PCONV
37 00265'000000 P0:0.
38 00266'000024 YOFST:20.
39 00267'000513'LN3AD:LINE3
40 00270'000551'LN0AD:LINE0
41 00271'000000 WDCTR:0
42 00272'177000 MSKHI:177000
43 00273'000377 MSKLO:377
44 00274'000000 LNCT:0
45                ;*************************************************************************
46                ;  ROUTINE TO OUTPUT A LINE OF 5A CHARACTERS TO THE TTY
47 00275'040777 LNOUT:STA 0,LNCT    ; AC0 CONTAINS LINE ADDRESS
48 00276'040773        STA 1,WDCTR
49 00277'024773        LDA 1,MSKHI
50 00300'030773        LDA 2,MSKLO
51 00301'022773        LDA 0,@LNCT
52 00302'107700        ANDS 0,1
53 00303'065111        DOAS 1,TTO
54 00304'103000 PLOP1:AND 2,0
55 00305'024765        LDA 1,MSKHI
56 00306'063611        SKPDN TTO
57 00307'000777        JMP .-1
58 00310'061111        DOAS 0,TTO
59 00311'010763        ISZ LNCT
```

```
  0005  LSXYZ
01 00312'010757          ISZ =0CTR
02 00313'000002          JMP .+2
03 00314'001400          JMP 0,3
04 00315'022757          LDA 0,=LNCT
05 00316'107700          ANDS 0,1
06 00317'063611          SKPDA TTO
07 00320'000777          JMP .-1
08 00321'063111          DOAS 1,TTO
09 00322'000762          JMP PLOP1
10                       ;  END OF OUTPUT ROUTINE
11                       ;*************************************************
12 00323'000000 RTRNP:0
13 00324'100000 MASK1:100000
14 00325'070000 MASK2:070000
15 00326'007000 MASK3:007000
16 00327'000700 MASK4:000700
17 00330'000070 MASK5:000070
18 00331'000007 MASK6:000007
19 00332'000000 TEMP:0
20 00333'000040 CODE:40
21                       ;*************************************************
22                       ;  ROUTINE TO CONVERT A 16 BIT WORD TO OCTAL CODE
23 00334'054747 PCONV:STA 3,RTRNP
24 00335'034004-         LDA 3,LMAP
25 00336'024746          LDA 1,MASK1
26 00337'107520          ANDZL 0,1
27 00340'125100          MOVL 1,1
28 00341'034772          LDA 2,CODE
29 00342'107300          ADDS 2,1
30 00343'044747          STA 1,TEMP
31 00344'024741          LDA 1,MASK2
32 00345'107720          ANDZS 0,1
33 00346'125200          MOVR 1,1
34 00347'125200          MOVR 1,1
35 00350'125200          MOVR 1,1
36 00351'125200          MOVR 1,1
37 00352'147000          ADD 2,1
38 00353'030757          LDA 2,TEMP
39 00354'147000          ADD 2,1
40 00355'045400          STA 1,0,3
41 00356'175400          INC 3,3
42 00357'024747          LDA 1,MASK3
43 00360'107720          ANDZS 0,1
44 00361'125200          MOVR 1,1
45 00362'030751          LDA 2,CODE
46 00363'147300          ADDS 2,1
47 00364'044746          STA 1,TEMP
48 00365'024742          LDA 1,MASK4
49 00366'107520          ANDZL 0,1
50 00367'125100          MOVL 1,1
51 00370'125300          MOVS 1,1
52 00371'147000          ADD 2,1
53 00372'030740          LDA 2,TEMP
54 00373'147000          ADD 2,1
55 00374'045400          STA 1,0,3
56 00375'175400          INC 3,3
57 00376'024732          LDA 1,MASK5
58 00377'107620          ANDZR 0,1
59 00400'125200          MOVR 1,1
```

```
  0006  LSXYZ
01 00401'125200        MOVR 1,1
02 00402'030731        LDA 2,CODE
03 00403'147300        ADDS 2,1
04 00404'044726        STA 1,TEMP
05 00405'024724        LDA 1,MASK6
06 00406'107400        AND 0,1
07 00407'147000        ADD 2,1
08 00410'030722        LDA 2,TEMP
09 00411'147000        ADD 2,1
10 00412'045000        STA 1,0,3
11 00413'175400        INC 3,3
12 00414'054004-       STA 3,LNAD
13 00415'000401        JMP .+1
14 00416'002705        JMP @RTRND
15                ;  END OF ROUTINE TO CONVERT A 16 BIT WORD TO OCTAL CODE
16                ;*********************************************************************
17                ;*********************************************************************
18                LINE1: .TXT /      X TT                7 TT                X DATA    <15><12>/
   00417'020040
   00420'020040
   00421'020040
   00422'054040
   00423'052111
   00424'020040
   00425'020040
   00426'020040
   00427'020040
   00430'020040
   00431'020040
   00432'020040
   00433'020040
   00434'055040
   00435'052111
   00436'020040
   00437'020040
   00440'020040
   00441'020040
   00442'020040
   00443'020040
   00444'020040
   00445'020040
   00446'054440
   00447'042101
   00450'052101
   00451'020040
   00452'020040
   00453'006415
   00454'000000
19                LINE2: .TXT /000000    000000    000000    000000    000000    000000<15><12>/
   00455'030060
   00456'030060
   00457'030060
   00460'020040
   00461'020040
   00462'030060
   00463'030060
   00464'030060
   00465'020040
   00466'020040
```

```
0007  LSAAZ
 00067'030060
 00070'030060
 00071'030060
 00072'020040
 00073'020040
 00074'030060
 00075'030060
 00076'030060
 00077'020040
 00500'020040
 00501'030060
 00502'030060
 00503'030060
 00504'020040
 00505'020040
 00506'030060
 00507'030060
 00510'030060
 00511'006412
 00512'000000
01                      ; LINE 2 IS TO BE USED WHEN X II, 7 II & Y AREAS HAVE DATA
02              LINE2: .TXT /000000    000000    000000    000000    000000    000000<15><12>/
 00513'030060
 00514'030060
 00515'030060
 00516'020040
 00517'020040
 00520'030060
 00521'030060
 00522'030060
 00523'020040
 00524'020040
 00525'030060
 00526'030060
 00527'030060
 00530'020040
 00531'020040
 00532'030060
 00533'030060
 00534'030060
 00535'020040
 00536'020040
 00537'030060
 00540'030060
 00541'030060
 00542'020040
 00543'020040
 00544'030060
 00545'030060
 00546'030060
 00547'006412
 00550'000000
03                      ; LINE 3 IS TO BE USED WHEN ONLY Y DATA REMAINS
04              LINE4: .TXT /000000    000000    000000    000000    000000    000000<15><12>/
 00551'030060
 00552'030060
 00553'030060
 00554'020040
 00555'020040
```

```
0008  LSXYZ
  00556'030040
  00557'030040
  00560'030040
  00561'020040
  00562'020040
  00563'030040
  00564'030040
  00565'030040
  00566'020040
  00567'020040
  00570'030040
  00571'030040
  00572'030040
  00573'020040
  00574'020040
  00575'030040
  00576'030040
  00577'030040
  00600'020040
  00601'020040
  00602'030060
  00603'030040
  00604'030040
  00605'004412
  00606'000000
01                      ; LINE 4  IS TO BE USED WHEN ONLY X,Z TI DATA REMAINS.
02              LINES: .TXT /NO TI OR Y DATA <15><12>/
  00607'047117
  00610'020124
  00611'044440
  00612'047522
  00613'020131
  00614'020104
  00615'040524
  00616'000440
  00617'004412
  00620'000000
03                    .END    ; END OF LSXYZ
```

```
 0009  LSXYZ

CHKY     000100'  2/27    2/36
CODE     000333'  5/20    5/28    5/45    6/02
CTRY     000004'  1/29    1/42    1/47    2/14
ENDXZ    000160'  2/04    2/12    2/25    2/57    3/25    4/03    4/23
ENDY     000161'  2/05    2/18    2/32    2/36    3/19    3/26    3/31    3/53
HAVOT    000035'  1/45    1/49    1/54
LINE1    000417'  1/30    6/18
LINE2    000455'  1/31    6/19
LINE3    000513'  4/39    7/02
LINE4    000551'  4/40    7/04
LINE5    000607'  1/32    8/02
LN1AD    000005'  1/30    2/20
LN2AD    000006'  1/31    2/23    3/21
LN2AR    000067'  2/23    3/24
LN3AD    000247'  3/29    3/55    4/39
LN3AR    000140'  2/39    3/29    3/58
LN4AD    000270'  4/01    4/28    4/40
LN4AR    000223'  2/34    4/01    4/32
LN5AD    000007'  1/32    1/50
LNAD     000004-  1/22    2/24    2/44    2/47    3/05    3/08    3/30    3/34
                  3/39    3/42    4/02    4/10    4/13    5/24    6/12
LNCT     000274'  4/44    4/47    4/51    4/59    5/04
LNOUT    000275'  1/33    3/23    3/57    4/30    4/47
LSXYZ    000011'  1/34
M29      000162'  2/21    3/22    3/27    3/56    4/29
MA       000163'  2/40    3/28    4/06
MB       000000'  1/25    1/51
MASK1    000324'  5/13    5/25
MASK2    000325'  5/14    5/31
MASK3    000326'  5/15    5/42
MASK4    000327'  5/16    5/48
MASK5    000330'  5/17    5/57
MASK6    000331'  5/18    6/05
MSKHI    000272'  4/42    4/49    4/55
MSKLO    000273'  4/43    4/50
NEGT1    000127'  2/54    2/59
NEGT2    000253'  4/20    4/25
PA       000265'  2/52    4/18    4/37
PCONV    000334'  3/45    4/36    5/23
PCTR     000243'  2/41    2/48    4/07    4/15    4/35
PLOP1    000304'  4/54    5/09
PLOP2    000106'  2/42    2/50
PLOP4    000232'  4/09    4/16
R2CT     000000-  1/18    1/57    2/42    2/48    4/08    4/14
RTRND    000001-  1/19    1/34    1/53    2/36    3/33    4/05
RTRNP    000323'  5/12    5/23    6/14
TEMP     000332'  5/19    5/30    5/38    5/47    5/53    6/04    6/08
TICTR    000003'  1/28    1/40    1/44    2/08
WDCTR    000271'  4/41    4/48    5/01
XZCTR    000002-  1/20    2/13    2/51    2/55    2/59    4/17    4/21    4/25
YCT      000005-  1/23    2/02    3/03    3/09    3/10    3/12    3/37    3/43
                  3/44    3/46
YCTR     000003-  1/21    2/19    3/13    3/16    3/47    3/50
YDEST    000266'  3/34    4/38
YOUT     000132'  2/58    3/03
.LNOT    000010'  1/33    1/52    2/22
.PCON    000264'  2/43    3/04    3/11    3/38    4/09    4/36
.R2      000001'  1/26    1/36    1/54
.Y       000002'  1/27    1/38    1/58
```

```
  0001  DRECM                    DRECM18A
01                                .TITL DRECM
02                                .ENT DRECM
03                  ;    DEMONSTRATION VERSION OF RECM PROGRAM, RECEIVING
04                  ;    DATA FROM THE PIM AND LEAKERS FROM ALL 4 CAMS,
05                  ;    AS WELL AS TRACK INITIATION DATA.
06                  ;    IT WILL CONTINUALLY SEND TO THE DISPLAY THE DATA
07                  ;    AND LEAKER COORDINATE POINTS.
08                  ;    IF KEY 0 IS SET, DISPLAY X TRACK INITIATION DATA
09                  ;    50 TIMES.
10                  ;    IF KEY 1 IS SET, DISPLAY Z TRACK INITIATION DATA
11                  ;    50 TIMES.
12                  ;    KEYS 2-8 WILL CONTAIN THE # OF TIMES THE DATA
13                  ;    LIST WILL BE DISPLAYED BEFORE CONTROL WILL
14                  ;    PASS TO THE LEAKER LIST.
15                  ;    KEYS 9-15 WILL CONTAIN THE # OF TIMES THE LEAKER
16                  ;    LIST WILL BE DISPLAYED BEFORE CONTROL WILL
17                  ;    PASS TO THE DATA LIST.
18                  ;    DESIGNED 16 JANUARY 1975
19                  ;    UPDATED 25 APRIL 1975 FOR TRACK INITIATION DATA
20                                .ZREL
21          000055                DISP=55    ; MNEUMONIC FOR DISPLAY
22 00000-176765 M523:-523.
23 00001-000000 CTR: 0
24 00002-000031 P25: 25.
25 00003-001040'.HEADR: HEADP
26 00004-001060'.PIMCT: PIMCT
27 00005-000000 .PIM:  0
28 00006-001061'.PIMHH: PIMHH
29 00007-001065'.CM3CT: CM3CT
30 00010-000000 .CM3LK:0
31 00011-001066'.CM3HH: CM3HH
32 00012-001072'.CM4CT: CM4CT
33 00013-000000 .CM4LK:0
34 00014-001073'.CM4HH: CM4HH
35 00015-001077'.CM5CT: CM5CT
36 00016-000000 .CM5LK:0
37 00017-001100'.CM5HH: CM5HH
38 00020-001104'.CM6CT: CM6CT
39 00021-000000 .CM6LK:0
40 00022-001105'.CM6HH: CM6HH
41 00023-000000 .LEAKR:0
42 00024-000000 LKCT:   0
43 00025-177776 M2: -2
44 00026-000000 WORD: 0
45 00027-000000 EVEN: 0
46 00030-177760 M16: -16.
47 00031-037600 MASKU: 037600
48 00032-000177 MASKL: 177
49 00033-140000 MSKTP:140000  ; MASK TO INDICATE DISPLAY TRACK INITIATION DATA ONLY
50 00034-100000 MSKTX:100000    ; MASK FOR X TRACK INITIATION DATA
51 00035-040000 MSKTZ:040000    ; MASK FOR Z TRACK INITIATION DATA
52 00036-001111'.TICT:TICT
53 00037-000000 .TIS:0
54 00040-000000 .TI:0
55 00041-001112'.TIMH:TIMH
56 00042-000000 .YS:0
57 00043-000000 .Y:0
58 00044-001116'.YCT:YCT
59 00045-001117'.YMH:YMH
```

```
  0002  DPECM
01 00046-000000 DLCTR: 0    ; DATA LOOP COUNTER
02 00047-000000 LLCTR: 0    ; LEAKER LOOP COUNTER
03 00050-000000 DATCT: 0
04 00051-001776 MSKXY: 1776    ; MASK OF X&Y UPPER 9 BITS
05                ;    MSKXY WILL BE CHANGED TO 1774 WHEN FULL
06                ;    SCALE DATA IS USED
07 00052-000000 DDCNT: 0   ; DISPLAY DATA COUNT
08 00053-000000 LLCNT: 0   ; DISPLAY LEAKER COUNT
09 00054-000000 STATR: 0
10 00055-000010 TIMDT: 1812
11 00056-000001 CNTDR: 1815
12 00057-000000 RWDAD: 0
13 00060-000000 RWDCT: 0
14 00061-100004 MDCDE: -32764.
15 00062-100001 DTCDE: -32767.
16 00063-000000 ETRST: 0
17 00064-000000 BLKAD: 0    ; BLOCK ADDRESS
18 00065-000000 SAVE: 0
19 00066-000001 CNTAD: 0
20 00067-120000 MSKPM: 120000
21 00070-020000 MSKC3: 020000
22 00071-040000 MSKC4: 040000
23 00072-030000 MSKC5: 030000
24 00073-060000 MSKC6: 060000
25 00074-050000 MSKR1:050000
26 00075-100000 MSKII:100000
27 00076-000215',.DLOP:DLOOP
28 00077-000176',.KYS1:KEYS1
29 00100-000220',.KYS2:KEYS2
30 00101-000307',.FLGS:FLAGS
31 00102-000000 NEWPM:0
32 00103-177777 M1:-1
33 00104-000105-.ZERO:ZERO
34 00105-000000 ZERO:0
35 00106-000000 .PIMS:0
36 00107-000404 .MMAX:404
37 00110-177773 M5:-5
38 00111-000012 P10:10.
39 00112-000002 P2:2
40 00113-023420 PEMM:10000.
41 00114-000312 P202:202.
42 00115-002000 P1024:1024.
43 00116-000624 P404:404.
44 00117-000000 .CM3S:0
45 00120-000000 .CM4S:0
46 00121-000000 .CM5S:0
47 00122-000000 .CM6S:0
48 00123-001033',.READY:READY
49 00124-170000 MCAMK:170000
50 00125-000736',.TLOP:TLOOP
51 00126-000000 TRIND:0
52 00127-000000 XAREA:0
53 00130-000000 ZAREA:0
54 00131-000000 XLCTR:0
55 00132-000000 ZLCTR:0
56 00133-000754',.XLOP:XLOOP
57 00134-000760',.ZLOP:ZLOOP
58 00135-000000 EXPTI:0
59 00136-000547',.RDIS:REDIS
```

```
  0003  DRECM
01 00137-000555'.LSNM:LISNM
02 00140-000321'.CKMCA:CKMCA
03 00141-177716 M50:-50.
04 00142-000000 XZCT:0
05                .NREL
06              ;   THIS SECTION INITIALIZES THE DATA & LEAKER
07              ;   ARRAYS, AND ESTABLISHES ARBITRARY COUNTER VALUES
08              ;   SO THAT THE DISPLAY CAN BEGIN WHILE THE PROGRAM
09              ;   WAITS FOR DATA.
10              ;   CLEAR PIM & LEAKER BLOCKS
11 00000'060277 DRECM:NIOC CPU
12 00001'062477       DICC 0,CPU
13 00002'060207       NIOC MCAR    ; READ IN MCA CODE FOR THIS MACHINE
14 00003'034123-      LDA 3,.READY
15 00004'062407       DTC 0,MCAR
16 00005'024124-      LDA 1,MCAMK
17 00006'107400       AND 0,1
18 00007'044074-      STA 1,MSKR1   ; STORE MCA CODE IN CURRENT MCA VARIABLE
19 00010'045001       STA 1,1,3     ;  & READY BLOCK
20 00011'034107-      LDA 3,.NMAX   ; THIS SECTION ESTABLISHES ADDRESSES
21 00012'024112-      LDA 1,P2      ; OF BLOCKS OF DATA OUTSIDE OF PROGRAM
22 00013'031400       LDA 2,0,3     ; STORAGE.
23 00014'050106-      STA 2,.PIMS   ; ADDRESS OF PIM'S DATA CODE BLOCK
24 00015'133000       ADD 1,2
25 00016'050005-      STA 2,.PIM    ; ADDRESS OF 10,000 WORDS OF X,Y COORDINATE POINTS
26 00017'020113-      LDA 0,PFRAM
27 00020'113000       ADD 0,2
28 00021'050117-      STA 2,.CM3S   ; ADDRESS OF CAM 3'S DATA CODE BLOCK
29 00022'133000       ADD 1,2
30 00023'050010-      STA 2,.CM3LK  ; ADDRESS OF CAM 3'S LEAKER LIST
31 00024'020114-      LDA 0,R202
32 00025'113000       ADD 0,2
33 00026'050120-      STA 2,.CM4S   ; ADDRESS OF CAM 4'S DATA CODE BLOCK
34 00027'133000       ADD 1,2
35 00030'050013-      STA 2,.CM4LK  ; ADDRESS OF CAM 4'S LEAKER LIST
36 00031'113000       ADD 0,2
37 00032'050121-      STA 2,.CM5S   ; ADDRESS OF CAM 5'S DATA CODE BLOCK
38 00033'133000       ADD 1,2
39 00034'050016-      STA 2,.CM5LK  ; ADDRESS OF CAM 5'S LEAKER LIST
40 00035'113000       ADD 0,2
41 00036'050122-      STA 2,.CM6S   ; ADDRESS OF CAM 6'S DATA CODE BLOCK
42 00037'133000       ADD 1,2
43 00040'050021-      STA 2,.CM6LK  ; ADDRESS OF CAM 6'S LEAKER LIST
44 00041'113000       ADD 0,2
45 00042'050023-      STA 2,.LEAKR  ; ADDRESS OF DISPLAY LEAKER LIST
46 00043'020116-      LDA 0,R404
47 00044'113000       ADD 0,2
48 00045'050037-      STA 2,.TIS    ; ADDRESS OF TRACK INITIATION'S DATA CODE BLOCK
49 00046'133000       ADD 1,2
50 00047'050040-      STA 2,.TI     ; ADDRESS OF TRACK INITIATION'S DATA LIST
51 00050'020115-      LDA 0,P1024
52 00051'113000       ADD 0,2
53 00052'050042-      STA 2,.YS     ; ADDRESS OF Y AREA'S DATA CODE BLOCK
54 00053'133000       ADD 1,2
55 00054'050043-      STA 2,.Y      ; ADDRESS OF Y AREA'S DATA LIST
56 00055'020110-      LDA 0,M5      ; CLEAR OUT HEADER, COUNTS, & MINI HEADERS
57 00056'040001-      STA 0,CTR
58 00057'034111-      LDA 3,R10
59 00060'126400       SUB 1,1
```

```
  0004  DRECM
01 00061'030003-       LDA 2,.MEADR
02 00062'045000 LOOPA: STA 1,0,2
03 00063'045001        STA 1,1,2
04 00064'045002        STA 1,2,2
05 00065'045003        STA 1,3,2
06 00066'045004        STA 1,4,2
07 00067'045005        STA 1,5,2
08 00070'045006        STA 1,6,2
09 00071'045007        STA 1,7,2
10 00072'045010        STA 1,10,2
11 00073'045011        STA 1,11,2
12 00074'173000        ADD 3,2
13 00075'010001-       ISZ CTR
14 00076'000744        JMP LOOPA
15 00077'045000        STA 1,0,2     ; CLEAR LAST VALUE
16 00100'020000-       LDA 0,M523    ; NOW CLEAR THE STORAGE AREA BEGINNING WITH PIMS.
17                                   ; THERE ARE 13,058 WORDS THERE TO CLEAR, BUT FOR EASE,
18                                   ; 13,075 WORDS ARE BEING CLEARED.
19 00101'040001-       STA 0,CTR
20 00102'030002-       LDA 3,P2S
21 00103'126400        SUB 1,1
22 00104'030106-       LDA 2,.PIMS
23 00105'045000 LOOP:  STA 1,0,2
24 00106'045001        STA 1,1,2
25 00107'045002        STA 1,2,2
26 00110'045003        STA 1,3,2
27 00111'045004        STA 1,4,2
28 00112'045005        STA 1,5,2
29 00113'045006        STA 1,6,2
30 00114'045007        STA 1,7,2
31 00115'045010        STA 1,10,2
32 00116'045011        STA 1,11,2
33 00117'045012        STA 1,12,2
34 00120'045013        STA 1,13,2
35 00121'045014        STA 1,14,2
36 00122'045015        STA 1,15,2
37 00123'045016        STA 1,16,2
38 00124'045017        STA 1,17,2
39 00125'045020        STA 1,20,2
40 00126'045021        STA 1,21,2
41 00127'045022        STA 1,22,2
42 00130'045023        STA 1,23,2
43 00131'045024        STA 1,24,2
44 00132'045025        STA 1,25,2
45 00133'045026        STA 1,26,2
46 00134'045027        STA 1,27,2
47 00135'045030        STA 1,30,2
48 00136'173000        ADD 3,2
49 00137'010001-       ISZ CTR
50 00140'000745        JMP LOOP
51 00141'020025-       LDA 0,M2
52 00142'042004-       STA 0,@.PIMCT
53 00143'042007-       STA 0,@.CM3CT
54 00144'042012-       STA 0,@.CM4CT
55 00145'042015-       STA 0,@.CM5CT
56 00146'042020-       STA 0,@.CM6CT
57 00147'042036-       STA 0,@.TICT
58              ;   CLEAR DISPLAY & MCA LINES
59 00150'060255        NIOC DISP
```

```
 0005  0RECM
01 00151'063555        SKPBZ DISP
02 00152'000777        JMP .-1
03 00153'063755        SKPDZ DISP
04 00154'000777        JMP .-1
05 00155'060207        NIOC MCAR
06 00156'063507        SKPBZ MCAR
07 00157'000400        JMR .
08 00160'063707        SKPDZ MCAR
09 00161'000400        JMR .
10 00162'102400        SUB 0,0    ; IF HDRD=0, HEADER BLOCK IS EXPECTED
11 00163'040026-       STA 0,HDRD   ; IF HDRD=1, DATA BLOCK IS EXPECTED
12 00164'040063-       STA 0,FIRST   ; IF EVEN=0, DISPLAY PIM DATA
13                                   ; IF EVEN=1, DISPLAY LEAKER DATA
14                                   ; IF FIRST=0, MERGE 2 WORDS OF X,Y
15                                    ; PIM DATA INTO 1 Y-X WORD
16                                   ; IF FIRST=1, DISPLAY PIM DATA AS IS
17 00165'040125-       STA 0,TRIND   ; IF TRIND=0, NO TRACK INITIATION DATA IS BEING
18                                     ;   DISPLAYED.
19                                     ; IF TRIND=1, TRACK INITIATION DATA IS BEING
20                                     ;   DISPLAYED.
21 00166'040127-        STA 0,XAREA    ; IF XAREA=0, DO NOT DISPLAY TRACK INITIATION
22                                     ;   X AREA DATA.
23                                     ; IF XAREA=1, DISPLAY T.I. X AREA DATA.
24 00167'040130-       STA 0,ZAREA    ; IF ZAREA=0, DO NOT DISPLAY TRACK INITIATION Z
25                                     ;   AREA DATA.
26                                     ; IF ZAREA=1, DISPLAY T.I.Z AREA DATA.
27 00170'040135-       STA 0,EXPTI    ; IF EXPTI=0, NO DATA IS EXPECTED FROM
28                                     ;   TRACK INITIATION MACHINE.
29                                     ; IF EXPTI=1, 2 BLOCKS OF DATA ARE EXPECTED AFTER
30                                     ;   THIS HEADER
31 00171'040102-       STA 0,NEWPM
32                ;   SET UP MCA RECEIVE FOR FIRST HEADER
33 00172'020030-       LDA 0,M16
34 00173'024003-       LDA 1,.HEADP
35 00174'062007        DOB 0,MCAR
36 00175'065107        DOBS 1,MCAR
37                ;   READ IN KEYS TO FIND FREQUENCY RATE FOR DISPLAYING DATA
38                ;   AND LEAKERS
39 00176'060477 KEYS1: READS 0
40 00177'024033-       LDA 1,MSKTR  ; IF EITHER BITS 0 OR 1 ARE SET,
41 00200'107404        AND 0,1,SZR; DISPLAY TRACK INITIATION DATA ONLY.
42 00201'002125-       JMP @.TLOP
43 00202'024031-       LDA 1,MASKU
44 00203'107520        ANDZL 0,1
45 00204'125300        MOVS 1,1
46 00205'124400        NEG 1,1
47 00206'040046-       STA 1,DLCTP
48 00207'024032-       LDA 1,MASKL
49 00210'107400        AND 0,1
50 00211'124400        NEG 1,1
51 00212'040047-       STA 1,LLCTR
52 00213'102400        SUB 0,0
53 00214'040027-       STA 0,EVEN
54                ;   PREPARE PIM DATA FOR DISPLAY
55 00215'020046-DLOOP:LDA 0,DLCTR
56 00216'101004        MOV 0,0,SZR
57 00217'000406        JMP DLOP1
58 00220'020047-       LDA 0,LLCTP
59 00221'101004        MOV 0,0,SZR
```

```
  000A  DPECM
01 00222'002100-        JMP @.KYS2
02 00223'102000         ADC 0,0
03 00224'040046-        STA 0,DLCTP  ; FORCE PIM DATA TO DISPLAY ONCE
04 00225'022004-DLOP1:LDA 0,@.PIMCT
05 00226'101242         MOVOP 0,0,SZC
06 00227'000400         JMP .   ; COUNT SHOULD ALWAYS BE EVEN
07 00230'101005         MOV 0,0,SNR
08 00231'000407         JMP ZPIM
09 00232'000401         JMP .+1
10 00233'040001-        STA 0,CTP
11 00234'040052-        STA 0,DDCNT   ; DISPLAY DATA COUNT
12 00235'034005-        LDA 3,.PIM
13 00236'054050-       STA 3,DATCT
14 00237'000413         JMP FTIME
15 00240'102000 ZPIM: ADC 0,0   ; WHEN PIM COUNT = 0,
16 00241'040052-        STA 0,DDCNT   ; SET UP DISPLAY FOR
17 00242'034005-        LDA 3,.PIM    ; 1 ZERO WORD
18 00243'054050-        STA 3,DATCT
19 00244'102400         SUB 0,0
20 00245'041400         STA 0,0,3
21 00246'101400         INC 0,0
22 00247'040063-        STA 0,FIRST
23 00250'042004-        STA 0,@.PIMCT
24 00251'000432         JMP DSPLA
25                ;   CHECK IF THIS IS FIRST TIME THROUGH THIS
26                ;   LOGIC FOR THIS DATA
27 00252'020063-FTIME:LDA 0,FIRST
28 00253'101004         MOV 0,0,SZR
29 00254'000427         JMP DSPLA
30 00255'101400         INC 0,0
31 00256'040063-        STA 0,FIRST
32                ;   MERGE X & Y WORDS
33                ;   Y=BITS 0-7, X=BITS 8-15
34 00257'021400 COMP1:LDA 0,0,3
35 00260'024051-        LDA 1,MSKXY
36 00261'107620         ANDZR 0,1
37 00262'000401         JMP .+1  ; THIS INSTRUCTION WILL BE REPLACED
38                               ; BY MOVR 1,1 (125200) WHEN
39                               ; FULL SCALE DATA IS USED
40 00263'175400         INC 3,3
41 00264'021400         LDA 0,0,3
42 00265'030051-        LDA 2,MSKXY
43 00266'113420         ANDZR 0,2
44 00267'000401         JMP .+1  ; THIS INSTRUCTION WILL BE REPLACED
45                               ; BY MOVR 2,2 (151200) WHEN
46                               ; FULL SCALE DATA IS USED
47 00270'151300         MOVS 2,2
48 00271'133000         ADD 1,2
49 00272'052050-        STA 2,@DATCT
50 00273'000401         JMP .+1
51 00274'010050-        ISZ DATCT
52 00275'175400         INC 3,3
53 00276'010001-        ISZ CTP
54 00277'000760         JMP COMP1
55 00300'102520         SUBZL 0,0
56 00301'040063-        STA 0,FIRST
57 00302'000401         JMP DSPLA
58                ;   START DMA TRANSFER TO DISPLAY
59 00303'020052-DSPLA:LDA 0,DDCNT
```

```
  0007  DRECM
01 00304'024005-        LDA 1,.PIM
02 00305'062055         DOR 0,DISP
03 00306'065155         DOAS 1,DISP
04 00307'063655 FLAGS:SKPDN DISP
05 00310'000777         JMP .-1
06 00311'020103-        LDA 0,M1
07 00312'024104-        LDA 1,.ZERO
08 00313'062055         DOR 0,DISP
09 00314'065155         DOAS 1,DISP
10 00315'063655         SKPDN DISP
11 00316'000777         JMP .-1
12 00317'063607         SKPDN MCAR
13 00320'002136-        JMP @.RDIS
14                ;   DONE FLAG SET ON MCA RECEIVE.  PROCESS DATA
15                ;   RECEIVED AND SET UP NEXT MCA RECEIVE
16                ;   CHECK STATUS WORD
17 00321'060407 CKMCA:DIA 0,MCAR
18 00322'040057-        STA 0,RWDAD
19 00323'061407         DIR 0,MCAR
20 00324'040060-        STA 0,RWDCT
21 00325'062407         DIC 0,MCAR
22 00326'040054-        STA 0,STATR
23 00327'024055-        LDA 1,TIMOT
24 00330'107404         AND 0,1,SZR
25 00331'000400         JMP .           ; RCVR TIME OUT
26 00332'024056-        LDA 1,CNTDR
27 00333'107405         AND 0,1,SNR
28 00334'000400         JMP .   ; RCVR COUNT NOT DONE
29 00335'000401         JMP .+1
30 00336'020024-        LDA 0,WORD
31 00337'101004         MOV 0,0,SZR
32 00340'000566         JMP DTBLK
33 00341'034003-        LDA 3,.HEADR
34 00342'021400 REPT1:LDA 0,0,3       ; CONTROL WILL PASS HERE IF EXPECTED DATA BLOCK WAS
35 00343'024061-        LDA 1,WDCDE     ; INCORRECT CODE
36 00344'106404         SUB 0,1,SZR
37 00345'002137-        JMP @.LSNW
38 00346'021401         LDA 0,1,3
39 00347'024067-        LDA 1,MSKPM
40 00350'106405         SUB 0,1,SNR
41 00351'000516         JMP SETPM
42 00352'024070-        LDA 1,MSKC3
43 00353'106404         SUB 0,1,SZR
44 00354'000405         JMP CHKC4
45 00355'030011-        LDA 2,.CM3MM
46 00356'020007-        LDA 0,.CM3CT
47 00357'024010-        LDA 1,.CM3LK
48 00360'000425         JMP SETCM
49 00361'024071-CHKC4:LDA 1,MSKC4
50 00362'106404         SUB 0,1,SZR
51 00363'000405         JMP CHKC5
52 00364'030014-        LDA 2,.CM4MM
53 00365'020012-        LDA 0,.CM4CT
54 00366'024013-        LDA 1,.CM4LK
55 00367'000416         JMP SETCM
56 00370'024072-CHKC5:LDA 1,MSKC5
57 00371'106404         SUB 0,1,SZR
58 00372'000405         JMP CHKC6
59 00373'030017-        LDA 2,.CM5MM
```

```
  000A  DRECM
01 00374'020015-        LDA 0,.CM5CT
02 00375'024016-        LDA 1,.CM5LK
03 00376'000407         JMP SETCM
04 00377'024073-CHKC6:LDA 1,MSKC6
05 00400'106404         SUB 0,1,SZR
06 00401'000431         JMP CHKTI
07 00402'030022-        LDA 2,.CM6MH
08 00403'020020-        LDA 0,.CM6CT
09 00404'024021-        LDA 1,.CM6LK
10 00405'040066-SETCM:STA 0,CNTAD
11 00406'040068-        STA 1,BLKAD
12 00407'021405         LDA 0,5,3
13 00410'041000         STA 0,0,2
14 00411'021406         LDA 0,6,3
15 00412'041001         STA 0,1,2
16 00413'021411         LDA 0,11,3
17 00414'041002         STA 0,2,2
18 00415'021412         LDA 0,12,3
19 00416'041003         STA 0,3,2
20 00417'021402         LDA 0,2,3
21 00420'030066-        LDA 2,CNTAD
22 00421'041000         STA 0,0,2
23 00422'000401         JMP .+1
24 00423'030025-        LDA 2,M2
25 00424'143000         ADD 2,0
26 00425'147000         ADD 2,1
27 00426'062007         DOR 0,MCAR
28 00427'065107         DOAS 1,MCAR
29 00430'010026-        ISZ WORD
30 00431'000656         JMP FLAGS
31 00432'024075-CHKTI:LDA 1,MSKTI
32 00433'106404         SUB 0,1,SZR
33 00434'063077         HALT            ; INVALID MCA CODE
34 00435'126520         SUBZL 1,1
35 00436'040135-        STA 1,EXPTI     ; SET TO +1, INDICATING T.I. DATA IS EXPECTED NEXT
36 00437'030041-        LDA 2,.TIMH
37 00440'020034-        LDA 0,.TICT
38 00441'024040-        LDA 1,.TI
39 00442'040066-        STA 0,CNTAD
40 00443'040068-        STA 1,BLKAD
41 00444'021415         LDA 0,15,3      ; MINI HEADER OF T.I. CONTAINS:
42 00445'041000         STA 0,0,2       ;  1) FRAME # OF ALL X AREA DATA
43 00446'021416         LDA 0,16,3
44 00447'041001         STA 0,1,2       ;  2) FRAME # OF ALL Z AREA DATA
45 00450'021417         LDA 0,17,3
46 00451'042004-        STA 0,4,YCT
47 00452'041002         STA 0,2,2       ;  3) -WORD COUNT OF Y AREA DATA WHICH
48 00453'000401         JMP .+1         ;        FOLLOWS T.I. DATA
49                                      ; END OF T.I. MINI HEADER DESCRIPTION
50 00454'021402         LDA 0,2,3       ; - WORD COUNT OF T.I. DATA WHICH FOLLOWS THIS HEADER
51 00455'030066-        LDA 2,CNTAD
52 00456'041000         STA 0,0,2
53 00457'000401         JMP .+1
54 00460'030025-        LDA 2,M2
55 00461'143000         ADD 2,0
56 00462'147000         ADD 2,1
57 00463'062007         DOR 0,MCAR
58 00464'065107         DOAS 1,MCAR
59 00465'010026-        ISZ WORD
```

```
  0009  DRECM
01 00466'000621          JMP FLAGS
02 00467'021405 SETPM:LDA 0,5,3   ; SET UP MINI HEADER
03 00470'030006-         LDA 2,.PIMMH
04 00471'041000          STA 0,0,2
05 00472'021406          LDA 0,6,3
06 00473'041001          STA 0,1,2
07 00474'102400          SUB 0,0
08 00475'040063-         STA 0,FIRST
09                    ;  SUSPEND DISPLAY UNTIL NEW PIM DATA IS IN
10 00476'060255          NIOC DISP
11 00477'063755          SKPDZ DISP
12 00500'000777          JMP .-1
13 00501'020103-         LDA 0,M1
14 00502'024104-         LDA 1,.ZERO
15 00503'062055          DOR 0,DISP
16 00504'065155          DOAS 1,DISP
17 00505'063655          SKPDN DISP
18 00506'000777          JMP .-1
19 00507'102520          SUBZL 0,0
20 00510'040102-         STA 0,NEWPM
21 00511'021402          LDA 0,2,3
22 00512'042004-         STA 0,@.PIMCT
23 00513'024005-         LDA 1,.PIM
24 00514'040064-         STA 1,BLKAD
25 00515'030025-         LDA 2,M2
26 00516'103000          ADD 2,0
27 00517'103000          ADD 2,1
28 00520'062007          DOR 0,MCAR
29 00521'065107          DOAS 1,MCAR
30 00522'010026-         ISZ WORD
31 00523'063607          SKPDN MCAR
32 00524'000777          JMP .-1
33                    ;  WHEN NEW PIM DATA IS IN, SET UP NEXT
34                    ;  RECEIVE.  ONCE THAT IS FINISHED,
35                    ;  CONTROL WILL PASS TO REDIS.
36 00525'002140-         JMP @.CKMCA
37                    ;  CHECK IF THIS IS A DATA BLOCK
38 00526'102400 DTBLK:SUB 0,0
39 00527'040026-         STA 0,WORD
40 00530'030064-         LDA 3,BLKAD
41 00531'030025-         LDA 2,M2
42 00532'157000          ADD 2,3
43 00533'021400          LDA 0,0,3
44 00534'024062-         LDA 1,DTCDE
45 00535'106404          SUB 0,1,SZR
46 00536'000604          JMP REPT1
47                    ; CHECK TO SEE IF THIS DATA BLOCK IS FROM TRACK INITIATION
48                    ;  MACHINE.  IF IS IS, SET UP ANOTHER RECEIVE FOR DATA INTO Y AREA
48 00537'020135-         LDA 0,EXPTI    ; IS THIS T.I. DATA?
50 00540'101005          MOV 0,0,SNR
51 00541'000410          JMP LISNM      ; NO - GO TO LISTEN FOR NEXT HEADER
52 00542'102400          SUB 0,0
53 00543'040135-         STA 0,EXPTI    ; YES - SET UP ANOTHER DATA BLOCK
54 00544'030025-         LDA 2,M2
55 00545'022044-         LDA 0,@.YCT
56 00546'103000          ADD 2,0
57 00547'024043-         LDA 1,.Y
58 00550'107000          ADD 2,1
59 00551'062007          DOR 0,MCAR
```

```
  0010  DPECM
01 00552'065107          DOAS 1,MCAR
02 00553'010026-         ISZ MORD
03 00554'002101-         JMP @.FLGS
04                ;    SET UP MCA RECEIVE EOP NEXT HEADER
05 00555'060297 LISNR:NIOC MCAR    ; UNLOCK RECEIVER
06 00556'020030-         LDA 0,M16    ; CONTROL WILL PASS HERE
07 00557'024003-         LDA 1,.HEADP   ;  1) WHEN DATA BLOCK IS RECEIVED
08 00560'062007          DOB 0,MCAP     ;      SUCCESSFULLY
09 00561'065107          DOAS 1,MCAR    ;  2) WHEN HEADER BLOCK IS NOT
10 00562'020102-         LDA 0,NEWPM
11 00563'101005          MOV 0,0,SNR
12 00564'002101-         JMP @.FLGS     ;      RECEIVED SUCCESSFULLY
13 00565'102400          SUB 0,0
14 00566'040102-         STA 0,NEWPM
15                ;    CONTROL PASSES TO PEDIS WHEN DISPLAY FLAG IS SET
16                ;    OR WHEN DISPLAY WAS TURNED OFF TO ACCEPT
17                ;    PIM DATA
18 00567'020126-PEDIS:LDA 0,TRIND    ; IS TRACK INITIATION DATA BEING DISPLAYED?
19 00570'101005          MOV 0,0,SNR
20 00571'000422          JMP LKPIM    ; NO
21 00572'020127-         LDA 0,XAREA  ; YES - IS X AREA DATA BEING DISPLAYED?
22 00573'101005          MOV 0,0,SNR
23 00574'000405          JMP .+5      ; NO
24 00575'010131-         ISZ XLCTR    ; YES
25 00576'002133-         JMP @.XLOP
26 00577'102400          SUB 0,0      ; CLEAR X AREA INDICATOR
27 00600'040127-         STA 0,XAREA
28 00601'020130-         LDA 0,ZAREA ; IS Z AREA BEING DISPLAYED?
29 00602'101005          MOV 0,0,SNR
30 00603'000405          JMP .+5   ; NO
31 00604'010132-         ISZ ZLCTR  ; YES
32 00605'002134-         JMP @.ZLOP
33 00606'102400          SUB 0,0     ; CLEAR Z AREA INDICATOR
34 00607'040130-         STA 0,ZAREA
35 00610'102400          SUB 0,0     ; CLEAR TRACK INITIATION INDICATOR
36 00611'040126-         STA 0,TPIND
37 00612'000413          JMP KEYS2
38 00613'020027-LKPIM:LDA 0,EVEN   ; WHAT IS BEING DISPLAYED?
39 00614'101004          MOV 0,0,SZR
40 00615'000404          JMP .+4    ;  LEAKERS ARE BEING DISPLAYED
41 00616'010045-         ISZ DLCTR    ; PIM DAT IS BEING DISPLAYED
42 00617'002076-         JMP @.DLOP    ; DISPLAY PIM DATA
43 00620'000404          JMP KEYS2
44 00621'010047-         ISZ LLCTR
45 00622'000421          JMP LLOOP    ; DISPLAY LEAKER DATA
46 00623'002077-         JMP @.KYS1
47 00624'060477 KEYS2:READS 0
48 00625'024033-         LDA 1,MSKIP
49 00626'107404          AND 0,1,SZR
50 00627'002125-         JMP @.TLOP
51 00630'024031-         LDA 1,MASKII
52 00631'107520          ANDZL 0,1
53 00632'125300          MOVS 1,1
54 00633'124400          NEG 1,1
55 00634'044046-         STA 1,DLCTR
56 00635'024032-         LDA 1,MASKL
57 00636'107400          AND 0,1
58 00637'124400          NEG 1,1
59 00640'044047-         STA 1,LLCTR
```

```
  0011  DREFM
01 00641'102520         SUBZL 0,0
02 00642'040027-        STA 0,EVEN
03                ;     PREPARE LEAKER DATA FOR DISPLAY
04 00643'020047-LLOOP:  LDA 0,LLCTR
05 00644'101004         MOV 0,0,SZR
06 00645'000402         JMP LLOP1
07 00646'002077-        JMP @.KYS1
08 00647'102400 LLOP1:  SUB 0,0
09 00650'040053-        STA 0,LLCNT
10 00651'030023-        LDA 2,.LEAKR
11 00652'050024-        STA 2,LKCT
12 00653'030010-        LDA 2,.CM3LK
13 00654'022007-        LDA 0,@.CM3CT
14 00655'101004         MOV 0,0,SZR
15 00656'004427         JSR MERGL
16 00657'030013-        LDA 2,.CM4LK
17 00660'022012-        LDA 0,@.CM4CT
18 00661'101004         MOV 0,0,SZR
19 00662'004423         JSR MERGL
20 00663'030016-        LDA 2,.CM5LK
21 00664'022015-        LDA 0,@.CM5CT
22 00665'101004         MOV 0,0,SZR
23 00666'004417         JSR MERGL
24 00667'030021-        LDA 2,.CMALK
25 00670'022020-        LDA 0,@.CMACT
26 00671'101004         MOV 0,0,SZR
27 00672'004413         JSR MERGL
28                ;     SEND MERGED LEAKER LIST TO DISPLAY
29 00673'020053-        LDA 0,LLCNT
30 00674'101004         MOV 0,0,SZR
31 00675'000404         JMP .+4
32 00676'034023-        LDA 3,.LEAKR    ; IF NO LEAKERS, FORCE DISPLAY
33 00677'041400         STA 0,0,3       ; OF 1 ZERO WORD
34 00700'102000         ADC 0,0
35 00701'024023-        LDA 1,.LEAKR
36 00702'062055         DOR 0,DISP
37 00703'065155         DOAS 1,DISP
38 00704'002101-        JMP @.FLGS
39 00705'050045-MERGL:  STA 3,SAVE
40 00706'101242         MOVOR 0,0,SZC
41 00707'000400         JMP .   ; COUNT SHOULD ALWAYS BE EVEN
42 00710'024053-        LDA 1,LLCNT
43 00711'107000         ADD 0,1
44 00712'044053-        STA 1,LLCNT
45 00713'040001-        STA 0,CTR  ; COUNT FOR THIS CAM
46 00714'021000 COMR2:  LDA 0,0,2
47 00715'024051-        LDA 1,MSKXY
48 00716'107620         ANDZR 0,1
49 00717'000401         JMP .+1  ; THIS INSTRUCTION WILL BE REPLACED
50                               ; BY MOVR 1,1 (125200) WHEN
51                               ; FULL SCALE DATA IS USED
52 00720'151400         INC 2,2
53 00721'021000         LDA 0,0,2
54 00722'034051-        LDA 3,MSKXY
55 00723'117620         ANDZR 0,3
56 00724'000401         JMP .+1  ; THIS INSTRUCTION WILL BE REPLACED
57                               ; BY MOVR 3,3 (175200) WHEN
58                               ; FULL SCALE DATA IS USED
59 00725'175300         MOVS 3,3
```

```
  0012  DWECM
01 00726'137000          ADD 1,3
02 00727'056020-         STA 3,@LKCT
03 00730'000401          JMP .+1
04 00731'010020-         ISZ LKCT
05 00732'151400          INC 2,2
06 00733'010001-         ISZ CTR
07 00734'000760          JMP COMP2
08 00735'002065-         JMP @SAVE
09 00736'126520 TLOOP:   SUBZL 1,1     ; SET TRACK INITIATION INDICATOR
10 00737'044126-         STA 1,TRIND
11 00740'024030-         LDA 1,MSKTX
12 00741'107405          AND 0,1,SNR
13 00742'000005          JMP CHCKZ
14 00743'126520          SUBZL 1,1
15 00744'044127-         STA 1,XAREA
16 00745'024141-         LDA 1,M50
17 00746'044131-         STA 1,XLCTR
18 00747'024035-CHCKZ:   LDA 1,MSKTZ
19 00750'107405          AND 0,1,SNR
20 00751'000005          JMP XLOOP
21 00752'024141-         LDA 1,M50
22 00753'044132-         STA 1,ZLCTR
23 00754'126520          SUBZL 1,1
24 00755'044130-         STA 1,ZAREA
25 00756'030040-XLOOP:   LDA 2,..TI    ; FIRST X AREA WORD
26 00757'000404          JMP SETTI
27 00760'030040-ZLOOP:   LDA 2,..TI
28 00761'151400          INC 2,2
29 00762'151400          INC 2,2       ; FIRST Z AREA WORD
30 00763'034023-SETTI:   LDA 3,..LEAKR  ; AC2 CONTAINS ADDRESS OF FIRST DATA WORD
31 00764'054020-         STA 3,LKCT     ; USE LEAKER AREA FOR STORAGE
32 00765'020524          LDA 0,TICT
33 00766'101005          MOV 0,0,SNR
34 00767'000404          JMP ZTI
35 00770'004412          ISP MRGTI
36 00771'020142-         LDA 0,YZCT     ; SEND MERGED TRACK INITIATION LIST TO DISPLAY
37                                      ;  OF EITHER X OR Z DATA
38 00772'000404          JMP .+4
39 00773'034023-ZTI:     LDA 3,..LEAKR  ; IF NO TRACKS, FORCE DISPLAY OF 1 ZERO WORD
40 00774'041403          STA 0,0,3
41 00775'102000          ADC 0,0        ; -1
42 00776'024023-         LDA 1,..LEAKR
43 00777'062055          IOR 0,DISP
44 01000'065155          IORS 1,DISP
45 01001'002101-         JMP @.FLGS
46 01002'054065-MRGTI:   STA 3,SAVE
47 01003'101242          MOVOR 0,0,SZC
48 01004'063077          HALT           ; COUNT SHOULD ALWAYS BE EVEN
49 01005'101240          MOVOR 0,0
50 01006'040001-         STA 0,CTR      ; ONLY  HALF THE DATA WILL BE PROCESSED HERE
51 01007'040142-         STA 0,XZCT     ;   EITHER X OR Z
52 01010'021000 SQWSH:   LDA 0,0,2
53 01011'024051-         LDA 1,MSKXY
54 01012'107620          ANDZR 0,1
55 01013'000401          JMP .+1        ; THIS INSTRUCTION WILL BE REPLACED BY
56                                      ; MOVP 1,1 (125200) WHEN FULL SCALE DATA IS USED.
57 01014'151400          INC 2,2
58 01015'021000          LDA 0,0,2
59 01016'034051-         LDA 3,MSKXY
```

```
  0013  DRECM
01 01017'117620          ANDZR 0,3
02 01020'000401          JMP .+1          ; THIS INSTRUCTION WILL BE REPLACED BY
03                                        ; MOVR 3,3 (175200) WHEN FULL SCALE DATA IS USED.
04 01021'175300          MOVS 3,3
05 01022'137000          ADD 1,3
06 01023'054024-         STA 3,@LKCT
07 01024'010021-         ISZ LKCT
08 01025'151400          INC 2,2
09 01026'151400          INC 2,2
10 01027'151400          INC 2,2
11 01030'010001-         ISZ CTR
12 01031'000757          JMP SQWSW
13 01032'002065-         JMP @SAVE
14 01033'100003 READY:-32765.
15 01034'000000          0
16 01035'000003          3    ; INDICATES DECM1 TYPE MACHINE
17 01036'000000          0
18 01037'000000          0
19       000040 HEADR: .BLK 40   ; GENERAL HEADER BLOCK
20 01060'000000 PICT:0    ; PIM COUNTER
21       000004 PIMH: .BLK 4    ; MINI HEADER FOR PIM
22 01065'000000 CM3CT:0   ; CAM 3 COUNTER
23       000004 CM3MH: .BLK 4      ; CAM3 MINI HEADER
24 01072'000000 CM4CT:0             ; CAM 4 COUNTER
25       000004 CM4MH: .BLK 4      ; CAM 4 MINI HEADER
26 01077'000000 CM5CT:0             ; CAM 5 COUNTER
27       000004 CM5MH: .BLK 4      ; CAM 5 MINI HEADER
28 01104'000000 CM6CT:0             ; CAM 6 COUNTER
29       000004 CM6MH: .BLK 4      ; CAM 6 MINI HEADER
30 01111'000000 TICT:0       ; TRACK INITIATION COUNTER
31       000004 TIMH: .BLK 4     ; TRACK INITIATION MINI HEADER
32 01116'000000 YCT:0               ; Y AREA COUNTER FROM TRACK INITIATION
33       000004 YMH: .BLK 4      ; Y AREA MINI HEADER FROM TRACK INITIATION
34       000001       .END DRECM
```

```
0014  DRECM

BLKAD   000044-     2/17    8/11    8/40    9/24    9/40
CHCK2   000747'    12/13   12/18
CHKCA   000341'     7/44    7/49
CHKCS   000370'     7/51    7/56
CHKCA   000377'     7/58    8/04
CHKTT   000432'     8/06    8/31
CKMCA   000321'     6/02    7/17
CM3CT   001045'     1/29   13/22
CM3MM   001066'     1/31   13/23
CM4CT   001072'     1/32   13/24
CM4MM   001073'     1/34   13/25
CM5CT   001077'     1/35   13/26
CM5MM   001100'     1/37   13/27
CM6CT   001104'     1/38   13/28
CM6MM   001105'     1/40   13/29
CNTAD   000046-     2/19    8/10    8/21    8/39    8/51
CNTDR   000056-     2/11    7/26
COMP1   000257'     6/34    6/54
COMP2   000714'    11/46   12/07
CTR     000001-     1/23    3/57    4/13    4/19    4/49    6/10    6/53   11/45
                   12/06   12/50   13/11
DATCT   000050-     2/03    6/13    6/18    6/49    6/51
DDCNT   000052-     2/07    6/11    6/16    6/59
DISP    000055      1/21    4/59    5/01    5/03    7/02    7/03    7/04    7/08
                    7/09    7/10    9/10    9/11    9/15    9/16    9/17   11/36
                   11/37   12/43   12/44
DLCTR   000046-     2/01    5/47    5/55    6/03   10/41   10/55
DLOOP   000215'     2/27    5/55
DLOP1   000225'     5/57    6/04
DRECM   000000'     3/11   13/34
DSPLA   000303'     6/24    6/29    6/57    6/59
DTALK   000526'     7/32    9/38
DTCDE   000042-     2/15    9/44
EVEN    000027-     1/45    5/53   10/38   11/02
EXPTI   000135-     2/58    5/27    8/35    9/49    9/53
FIRST   000043-     2/16    5/12    6/22    6/27    6/31    6/56    9/08
FLAGS   000307'     2/30    7/04    8/30    9/01
FTIME   000252'     6/14    6/27
HDCDE   000061-     2/14    7/35
HEADR   001040'     1/25   13/19
HORD    000026-     1/44    5/11    7/30    8/29    8/59    9/30    9/39   10/02
KEYS1   000176'     2/28    5/39
KEYS2   000624'     2/29   10/37   10/43   10/47
LTSNM   000555'     5/01    9/51   10/05
LKCT    000024-     1/42   11/11   12/02   12/04   12/31   13/06   13/07
LKPIM   000613'    10/20   10/38
LLCNT   000053-     2/08   11/09   11/29   11/42   11/44
LLCTR   000047-     2/02    5/51    5/58   10/44   10/59   11/04
LLOOP   000443'    10/35   11/04
LLOP1   000447'    11/06   11/08
LOOP    000105'     4/23    4/50
LOOPA   000042'     4/02    4/14
M1      000103-     2/32    7/06    9/13
M1A     000030-     1/46    5/33   10/06
M2      000025-     1/43    4/51    8/24    8/54    9/25    9/41    9/54
MS      000110-     2/37    3/56
MS0     000141-     3/03   12/16   12/21
MS23    000000-     1/22    4/16
MASKL   000032-     1/48    5/48   10/56
```

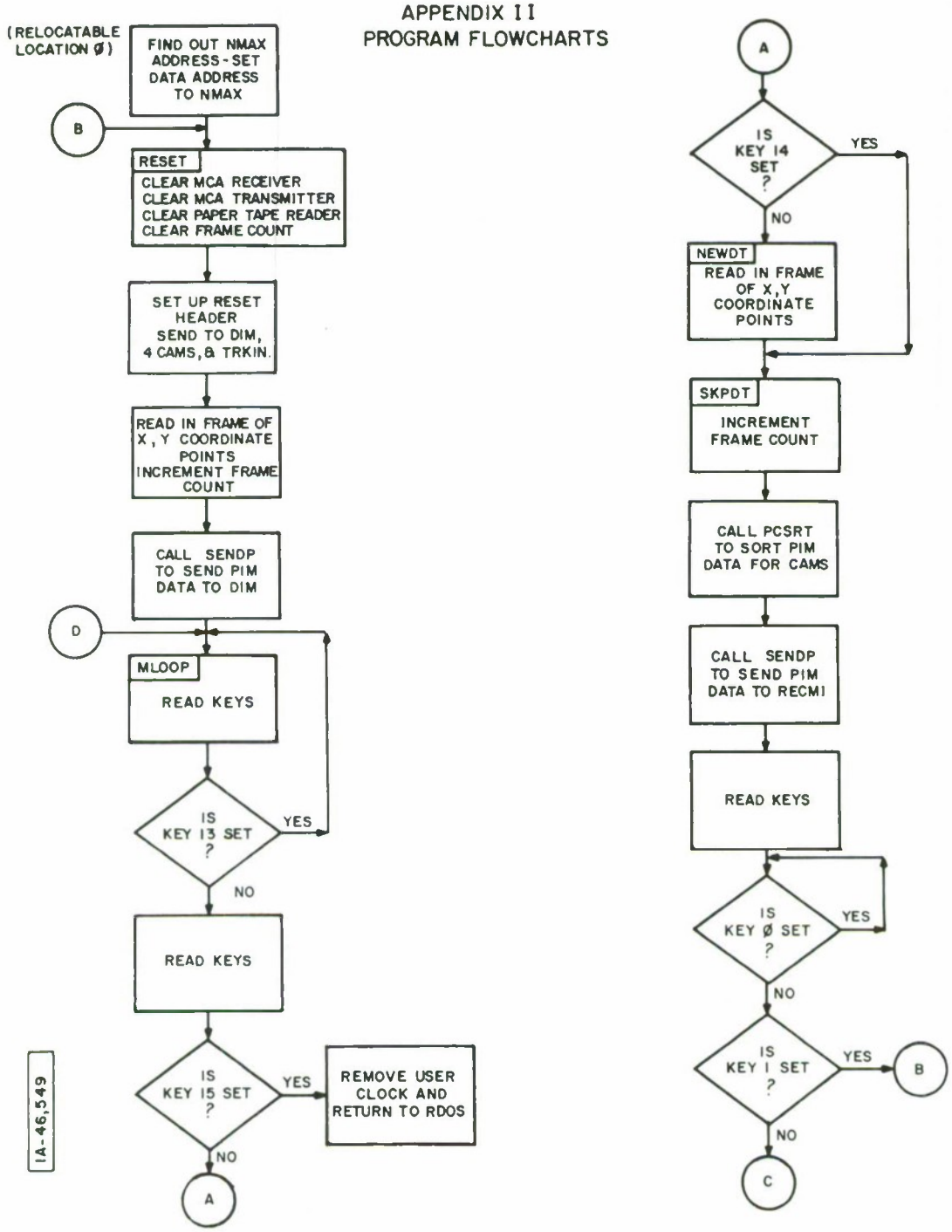
0015 DRECM

MASKU	000031-	1/47	5/43	10/51				
MCAMK	000124-	2/49	3/16					
MERGL	000705'	11/15	11/19	11/23	11/27	11/39		
MPGTI	001002'	12/35	12/44					
MSKC3	000070-	2/21	7/42					
MSKC4	000071-	2/22	7/49					
MSKC5	000072-	2/23	7/56					
MSKCA	000073-	2/24	8/04					
MSKPM	000067-	2/20	7/39					
MSKPI	000074-	2/25	3/1A					
MSKTI	000075-	2/26	8/31					
MSKTR	000033-	1/49	5/40	10/48				
MSKTX	000034-	1/50	12/11					
MSKTZ	000035-	1/51	12/1A					
MSKXY	000051-	2/04	6/35	6/42	11/47	11/54	12/53	12/59
NEWPM	000102-	2/31	5/31	9/20	10/10	10/14		
P10	000111-	2/38	3/58					
P1024	000115-	2/42	3/51					
P2	000112-	2/39	3/21					
P202	000114-	2/41	3/31					
P25	000002-	1/20	4/20					
R404	000116-	2/43	3/46					
PEPAM	000113-	2/40	3/26					
RIMCT	001000'	1/25	13/20					
PIMMM	001001'	1/28	13/21					
PEADY	001033'	2/48	13/14					
REDIS	000547'	2/59	10/1A					
PERT1	000342'	7/34	9/46					
RKDAD	000057-	2/17	7/1A					
RNDCT	000000-	2/13	7/20					
SAVE	000065-	2/1A	11/39	12/0A	12/46	13/13		
SETCM	000005'	7/48	7/55	8/03	8/10			
SETRM	000047'	7/41	9/02					
SETTI	000743'	12/2A	12/30					
SUBSM	001010'	12/52	13/12					
STATR	000050-	2/09	7/22					
TICT	001111'	1/52	12/32	13/30				
TIMM	001113'	1/55	13/31					
TIMOT	000055-	2/10	7/23					
TLOOP	000736'	2/50	12/09					
TRIMD	00012A-	2/51	5/17	10/1A	10/3A	12/10		
XARFA	000127-	2/52	5/21	10/21	10/27	12/15		
XLCTR	000131-	2/54	10/2A	12/17				
XLOOR	000756'	2/56	12/20	12/25				
XZCT	000102-	3/04	12/3A	12/51				
YCT	001116'	1/58	13/32					
YMM	001117'	1/59	13/33					
ZAREA	000130-	2/53	5/2A	10/2B	10/3A	12/2A		
ZERO	000105-	2/33	2/34					
ZLCTR	000132-	2/55	10/31	12/22				
ZLOOR	000760'	2/57	12/27					
ZRIM	000240'	6/0A	6/15					
ZTI	000773'	12/3A	12/39					
.CKMC	000140-	3/02	9/36					
.CM3C	000007-	1/29	4/53	7/46	11/13			
.CM3L	000010-	1/30	3/30	7/47	11/12			
.CM3M	000011-	1/31	7/45					
.CM3S	000117-	2/44	3/28					
.CM4C	000012-	1/32	4/54	7/53	11/17			

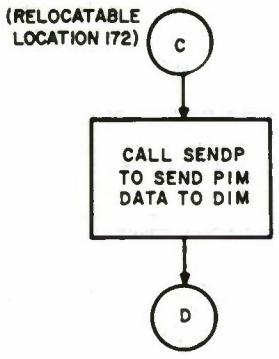
0016 DREDM

.CM4L	000013-	1/33	3/35	7/54	11/16				
.CM4M	000014-	1/34	7/52						
.CM4S	000120-	2/45	3/33						
.CM5C	000015-	1/35	4/55	A/01	11/21				
.CM5L	000016-	1/36	3/39	A/02	11/20				
.CM5M	000017-	1/37	7/50						
.CM5S	000121-	2/46	3/37						
.CMAC	000020-	1/38	4/5A	A/0A	11/25				
.CMAL	000021-	1/39	3/43	A/09	11/24				
.CMAM	000022-	1/40	A/07						
.CMAS	000122-	2/47	3/41						
.DLOP	00007A-	2/27	10/42						
.FLGS	000101-	2/30	10/03	10/12	11/3A	12/45			
.HEAD	000003-	1/25	4/01	5/34	7/33	10/07			
.KYS1	000077-	2/2A	10/46	11/07					
.KYS2	000100-	2/29	A/01						
.LEAK	000023-	1/41	3/45	11/10	11/32	11/35	12/30	12/39	12/42
.LSAM	000137-	3/01	7/37						
.MAX	000107-	2/36	3/20						
.PIM	000005-	1/27	3/25	6/12	6/17	7/01	9/23		
.PIMC	000004-	1/26	4/52	6/04	6/23	9/22			
.PIMM	00000A-	1/2A	9/03						
.PIMS	00010A-	2/35	3/23	4/22					
.PDTS	000136-	2/59	7/13						
.READ	000123-	2/48	3/14						
.TT	000040-	1/54	3/50	A/3A	12/25	12/27			
.TTCT	00003A-	1/52	4/57	A/37					
.TTMM	000041-	1/55	A/3A						
.TTS	000037-	1/53	3/4A						
.TLOP	000125-	2/50	5/42	10/50					
.XLOP	000133-	2/56	10/25						
.Y	000043-	1/57	3/55	9/57					
.YCT	000044-	1/58	A/4A	9/55					
.YMM	000045-	1/59							
.YS	000042-	1/56	3/53						
.ZAPO	000104-	2/33	7/07	9/14					
.ZLOP	000134-	2/57	10/32						

APPENDIX II
PROGRAM FLOWCHARTS

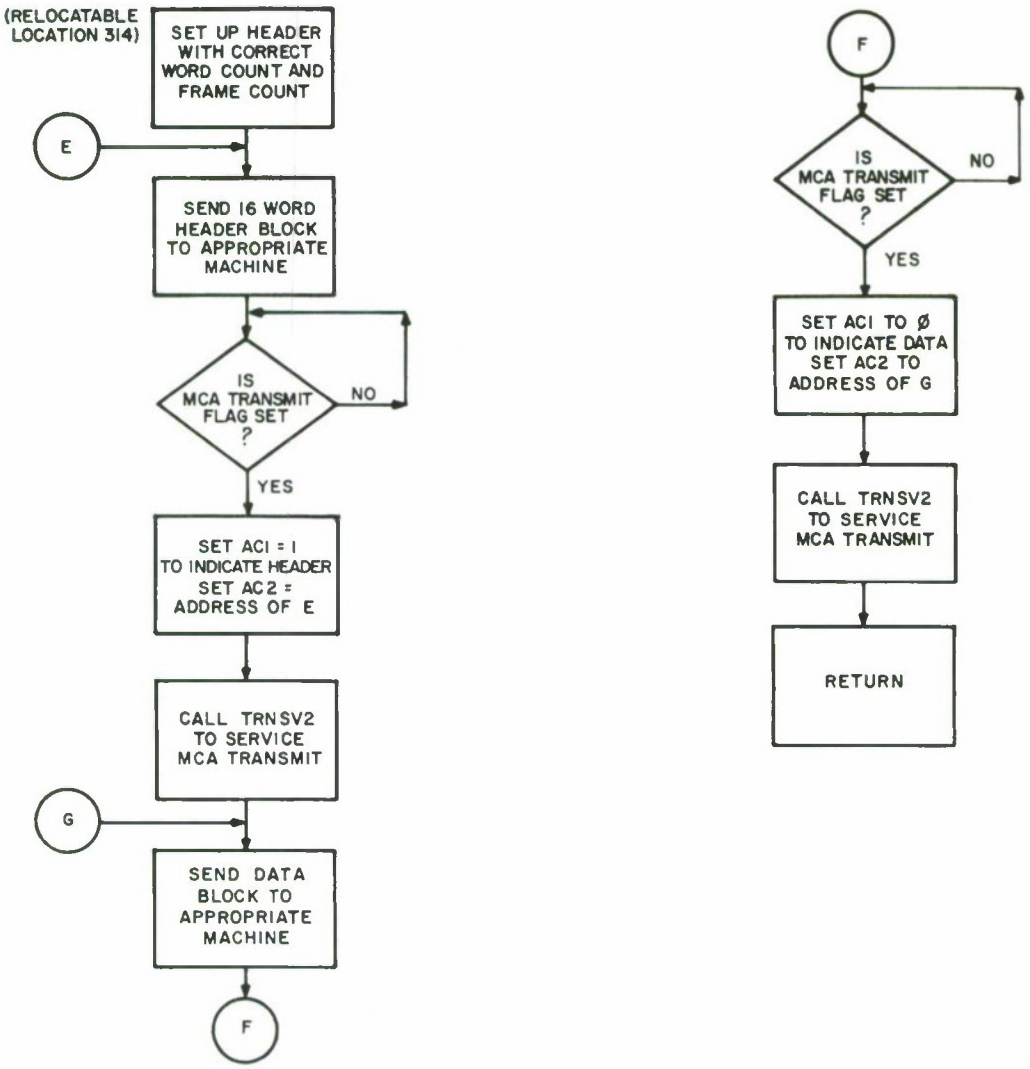


PIM PROGRAM – FOR DEMONSTRATION (LISTING STARTS ON PAGE 81)



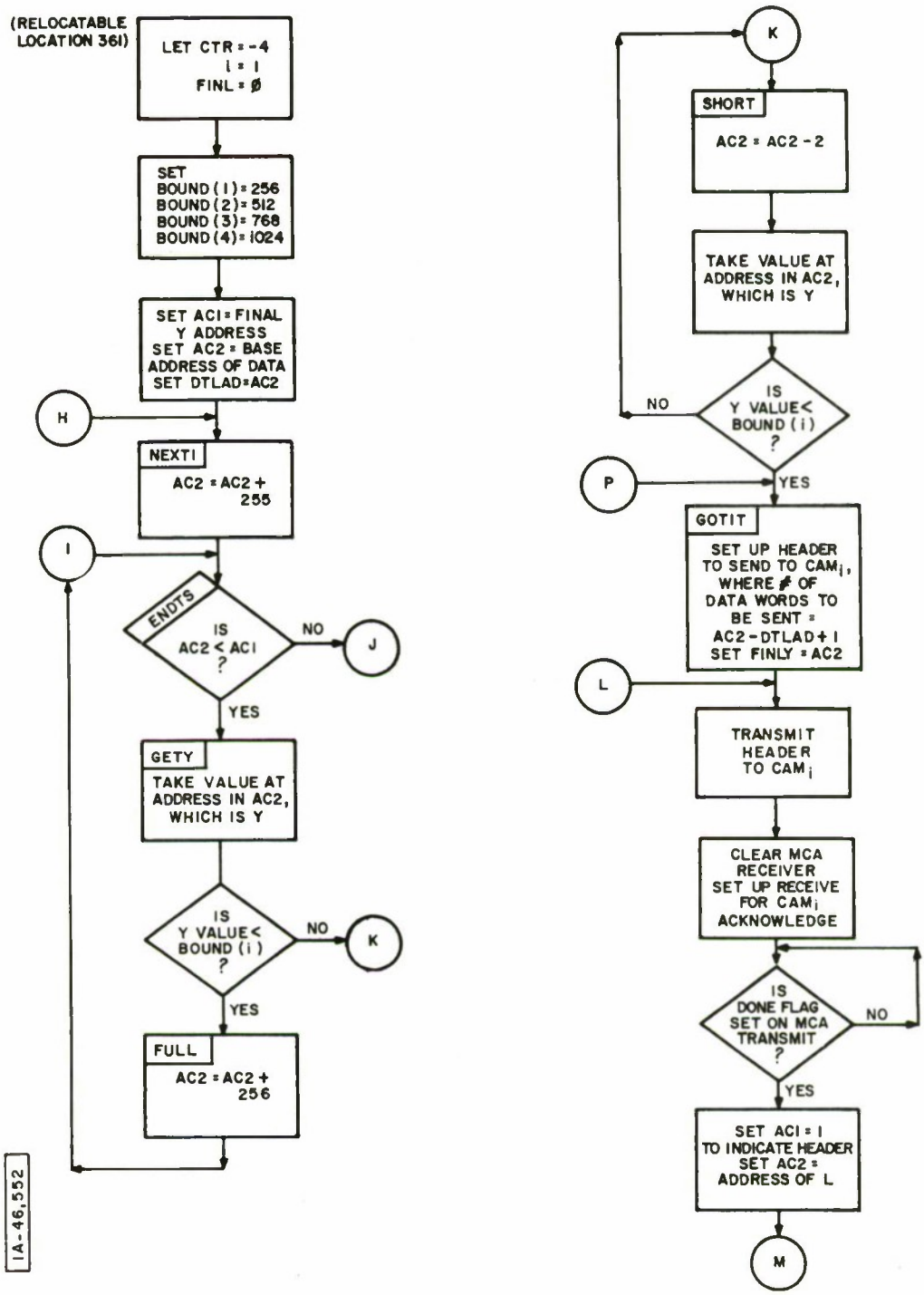
IA-46,550

PIM - 2

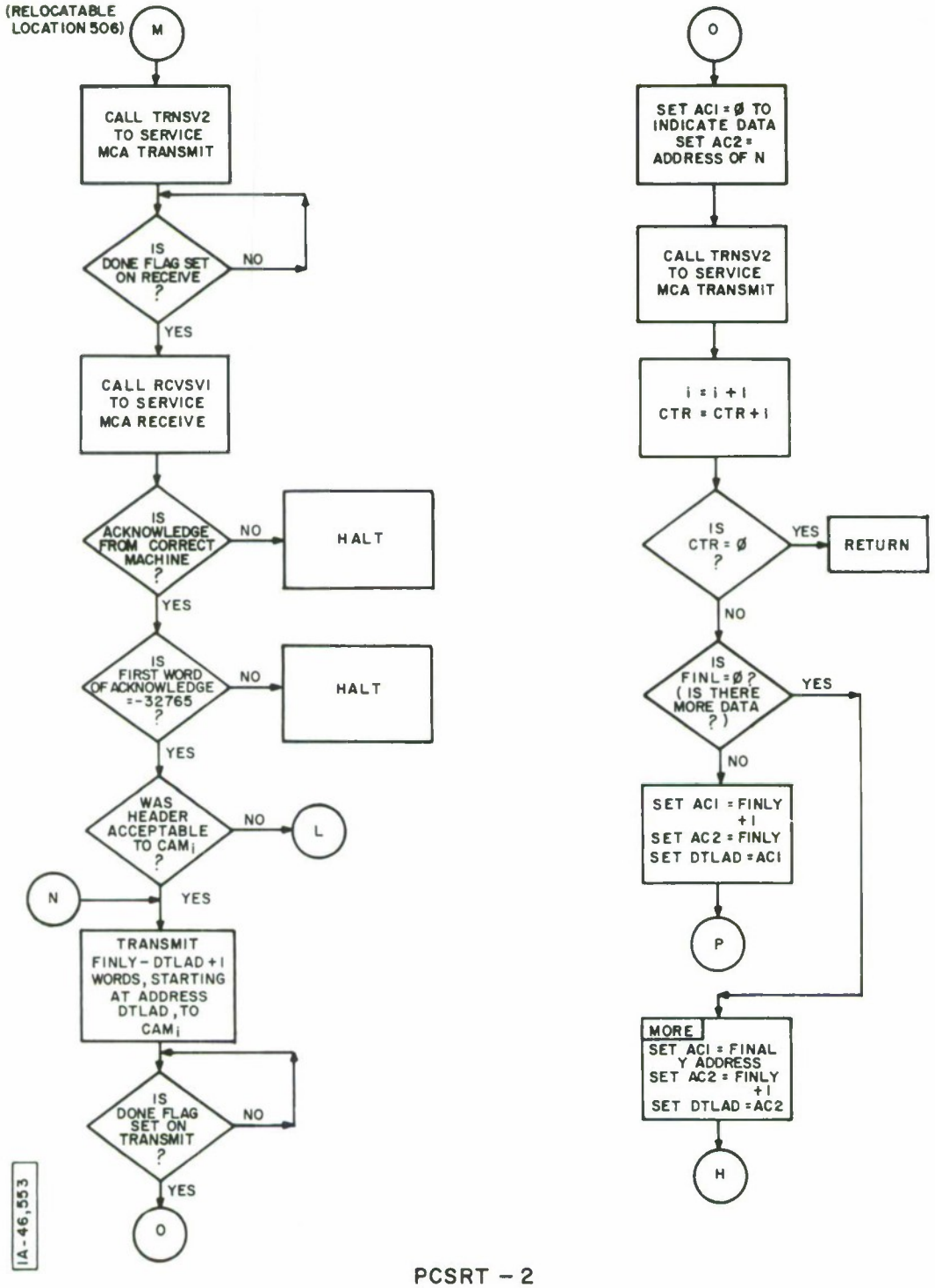


IA-46,551

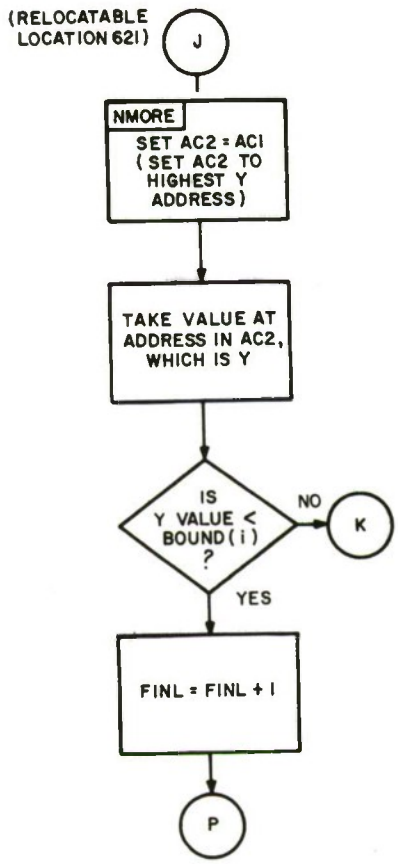
SENDP – SUBROUTINE OF PIM (LISTING STARTS ON PAGE 87)



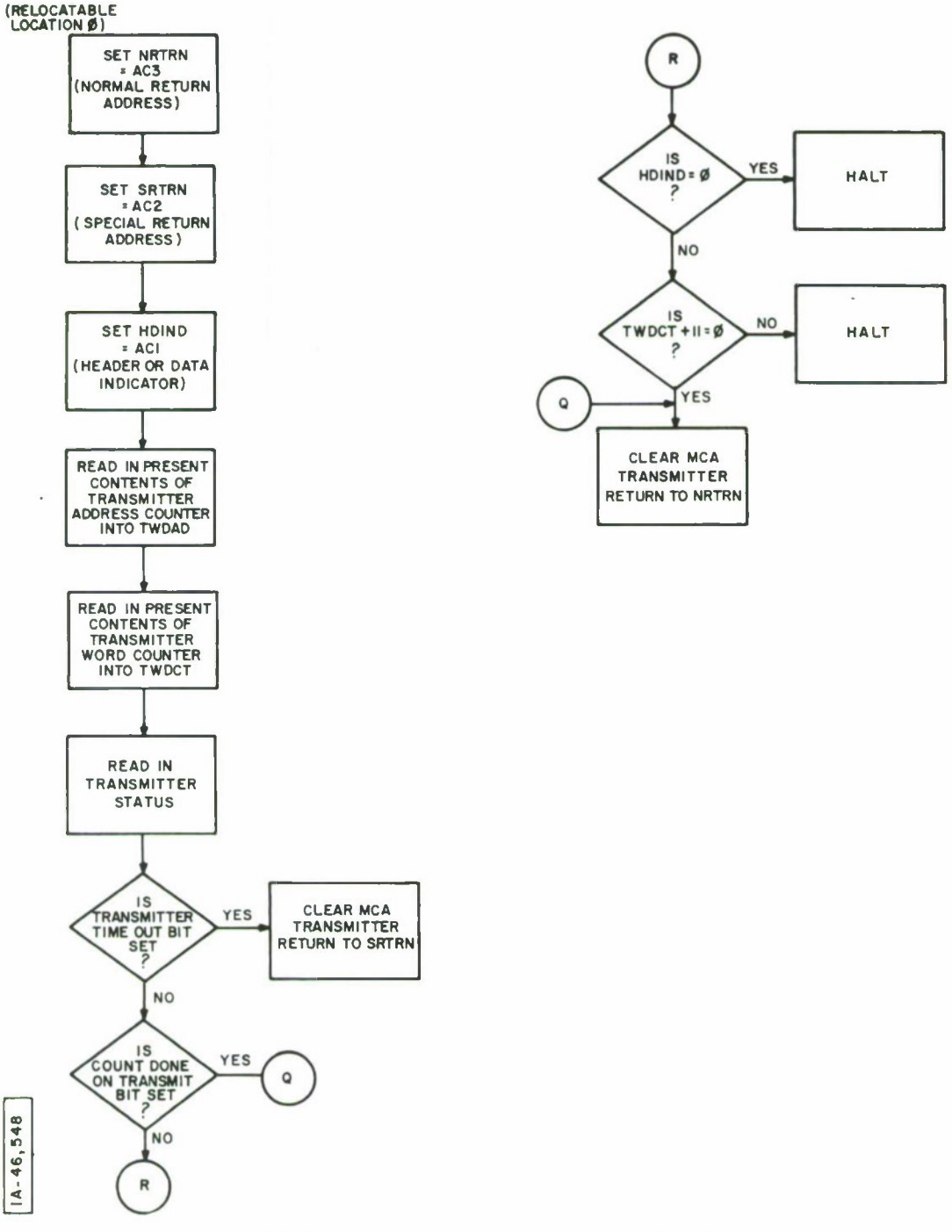
PCSRT – SUBROUTINE OF PIM (LISTING STARTS ON PAGE 88)


(RELOCATABLE LOCATION 506)
M
CALL TRNSV2 TO SERVICE MCA TRANSMIT
IS DONE FLAG SET ON RECEIVE ?
NO
YES
CALL RCVSVI TO SERVICE MCA RECEIVE
IS ACKNOWLEDGE FROM CORRECT MACHINE ?
NO
HALT
YES
IS FIRST WORD OF ACKNOWLEDGE =-32765 ?
NO
HALT
YES
WAS HEADER ACCEPTABLE TO CAMi ?
NO
L
N
YES
TRANSMIT FINLY-DTLAD+1 WORDS, STARTING AT ADDRESS DTLAD, TO CAMi
IS DONE FLAG SET ON TRANSMIT ?
NO
YES
O
IA-46,553
O
SET AC1=Ø TO INDICATE DATA SET AC2= ADDRESS OF N
CALL TRNSV2 TO SERVICE MCA TRANSMIT
i=i+1 CTR=CTR+1
IS CTR=Ø ?
YES
RETURN
NO
IS FINL=Ø? (IS THERE MORE DATA ?)
YES
NO
SET AC1=FINLY+1 SET AC2=FINLY SET DTLAD=AC1
P
MORE
SET AC1=FINAL Y ADDRESS SET AC2=FINLY+1 SET DTLAD=AC2
H

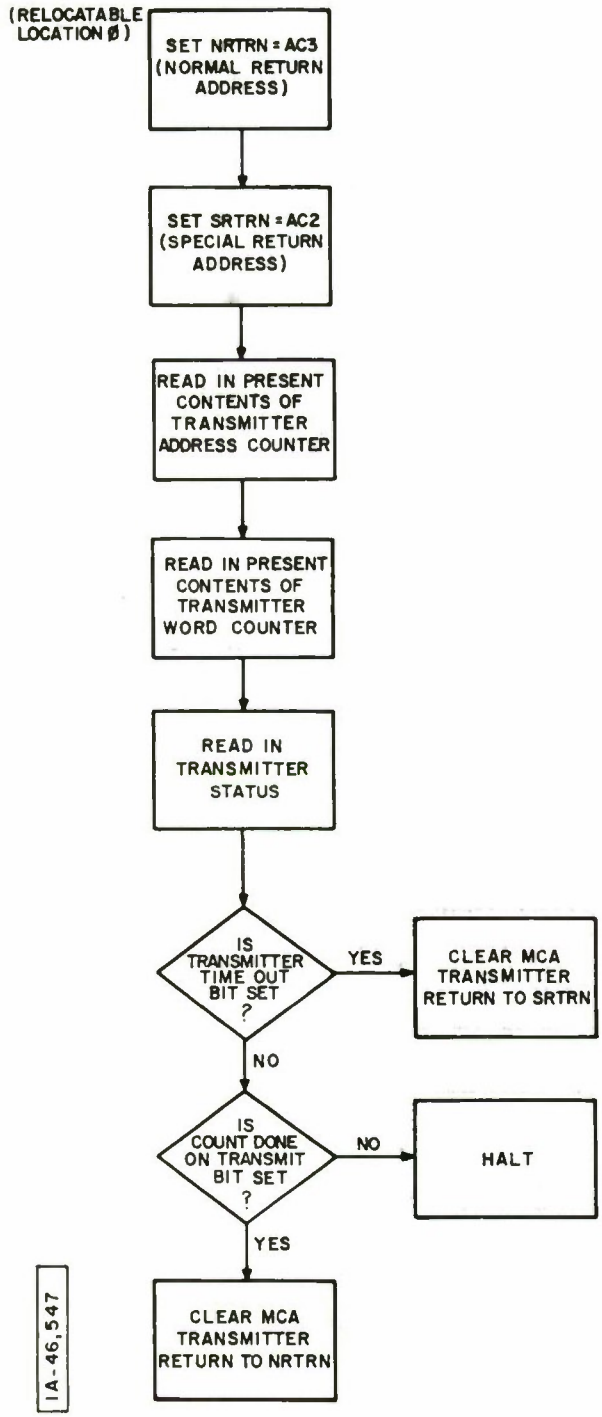
PCSRT – 2


(RELOCATABLE LOCATION 621)
J
NMORE
SET AC2 = AC1 (SET AC2 TO HIGHEST Y ADDRESS)
TAKE VALUE AT ADDRESS IN AC2, WHICH IS Y
IS Y VALUE < BOUND(i) ?
NO
K
YES
FINL = FINL + 1
P

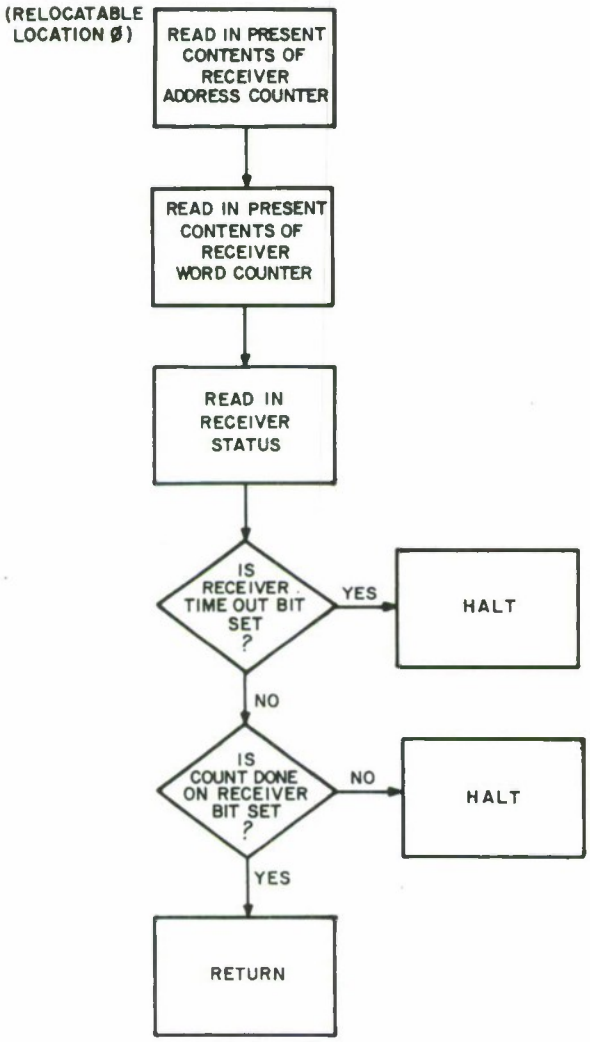
IA-46,554



TRNSV2 - SUBROUTINE OF PIM AND DIM (LISTING ON PAGE 94)

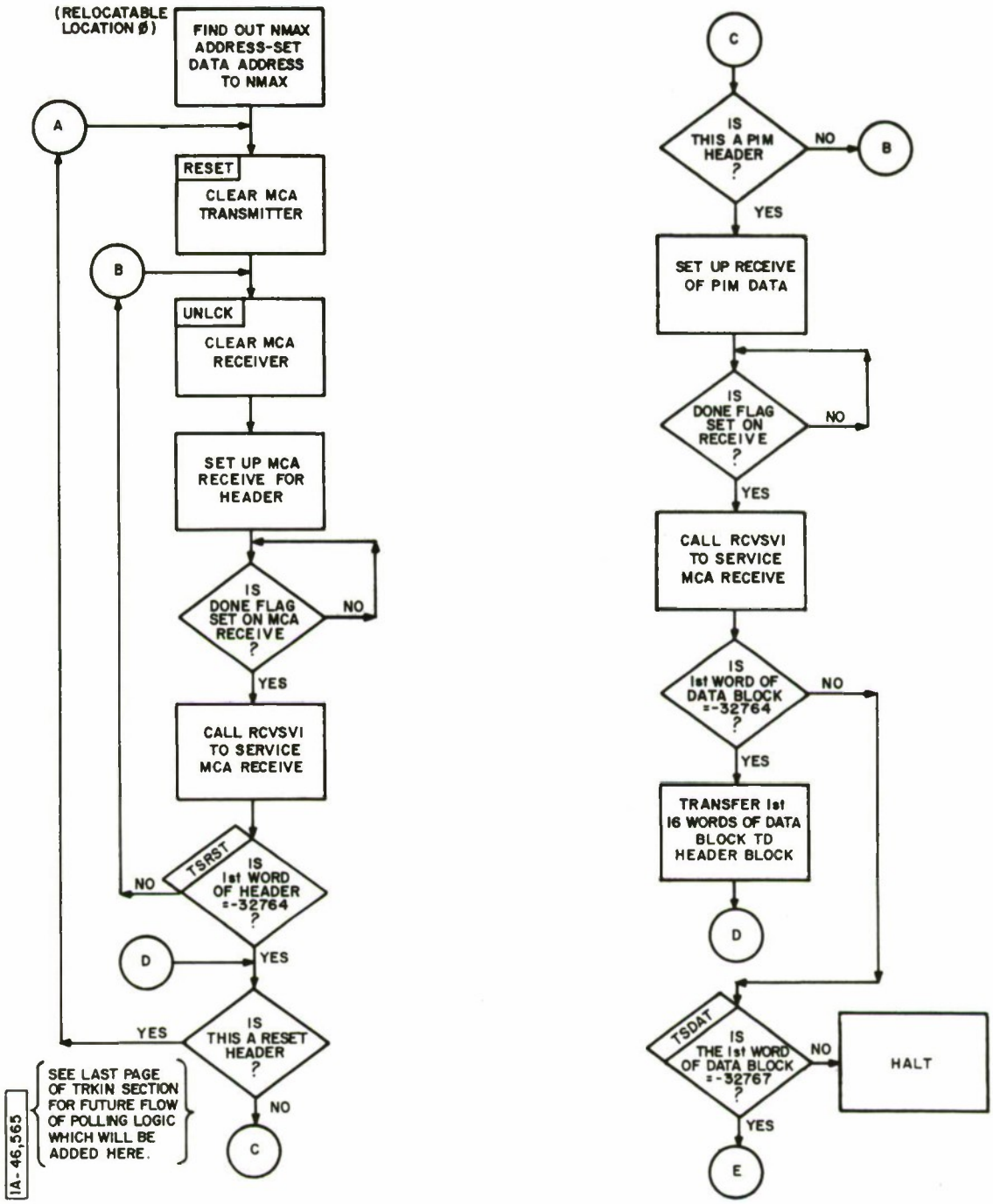


TRNSVI - SUBROUTINE OF CAM AND TRKIN (LISTING ON PAGE 95)

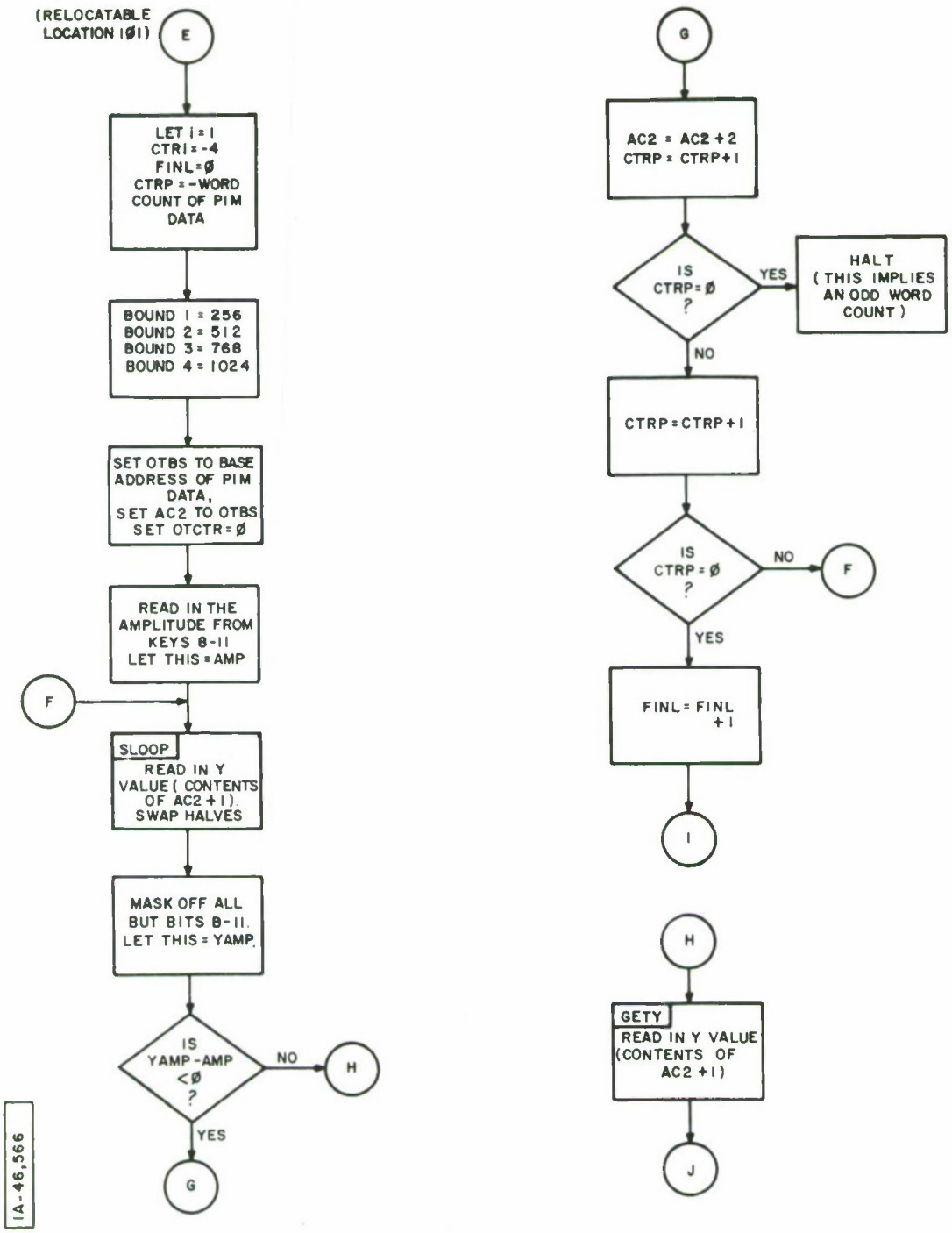


IA-46,546

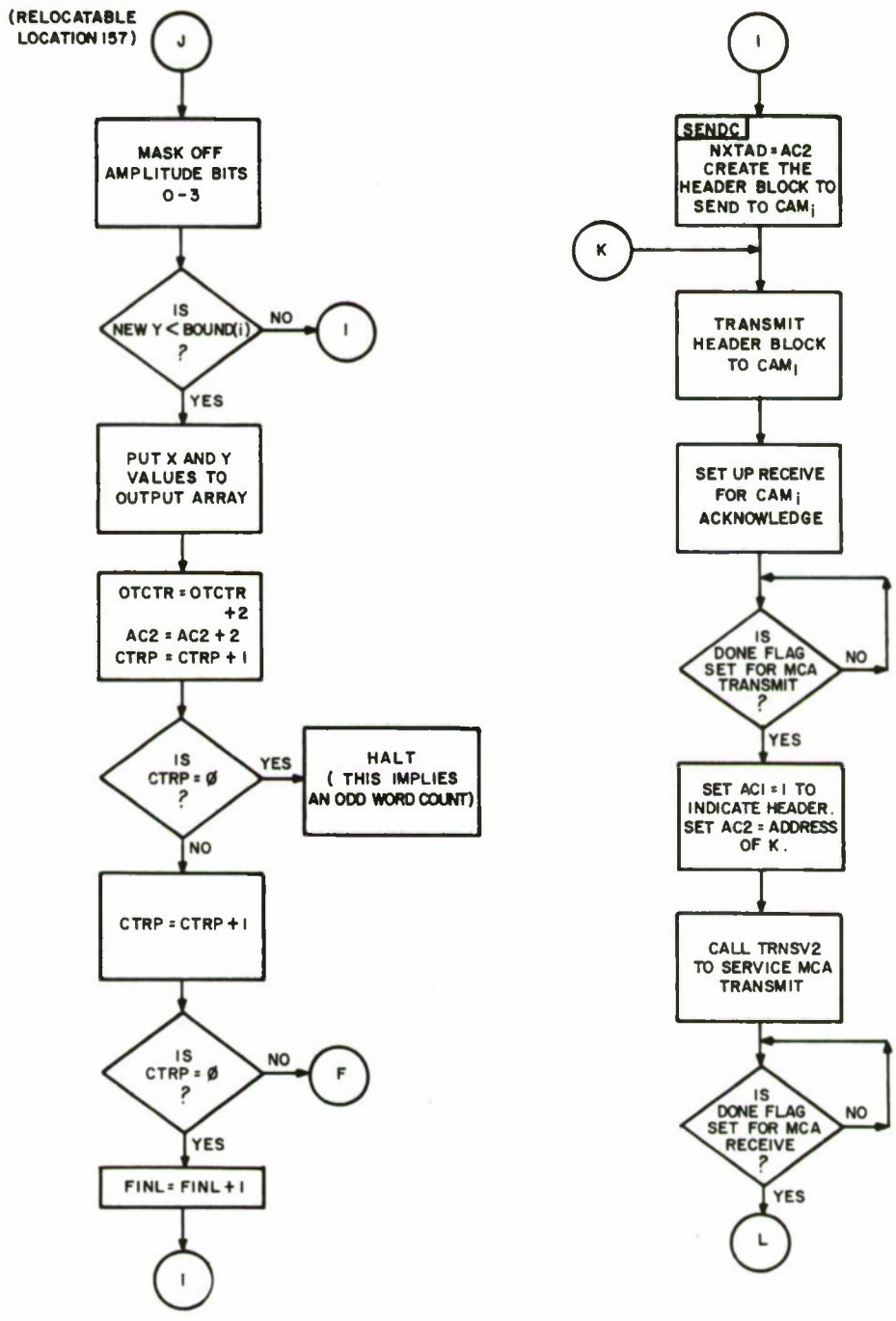
RCVSVI - SUBROUTINE OF PIM, DIM, CAM, AND TRKIN (LISTING ON PAGE 96)



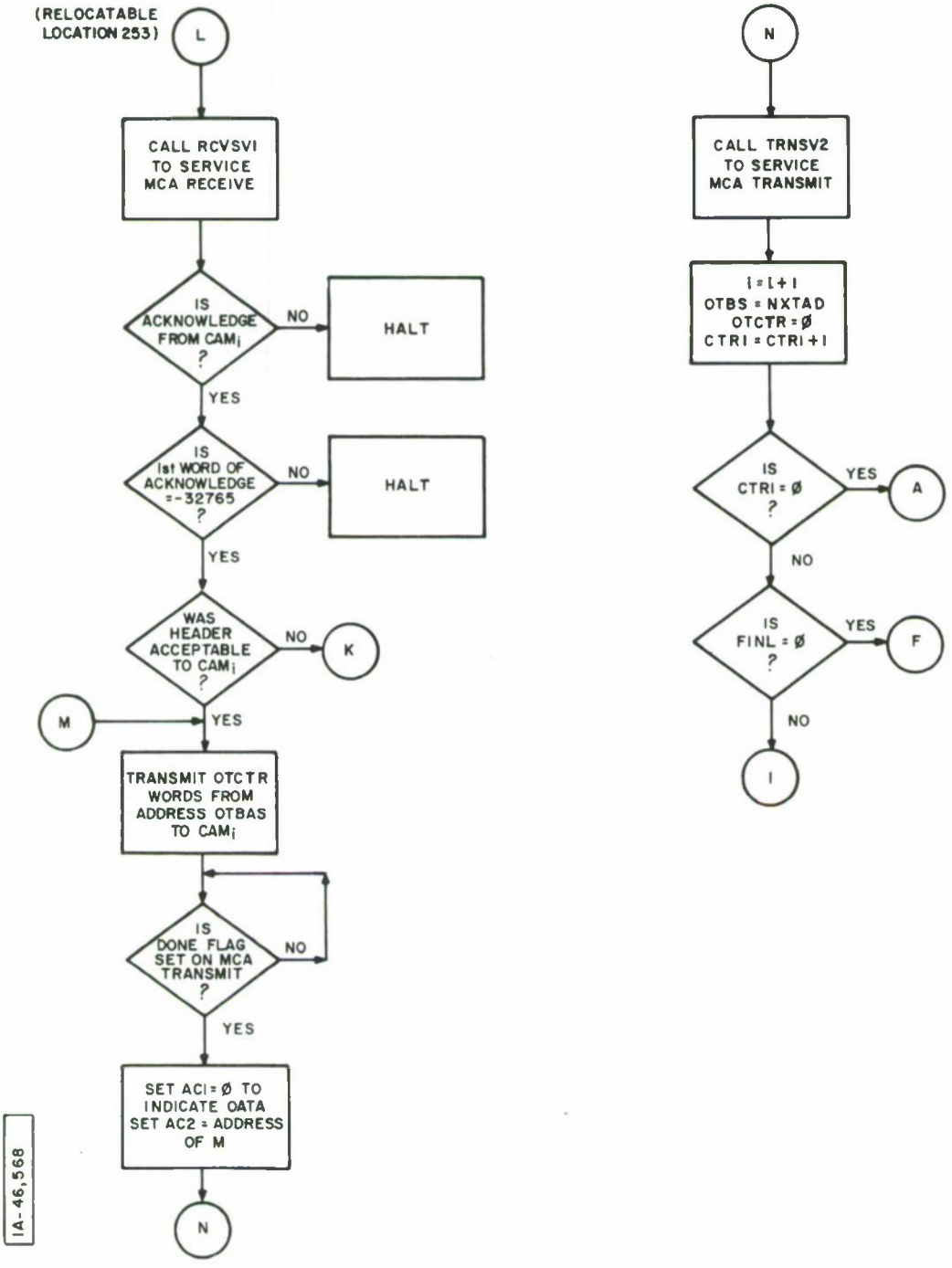
DIM PROGRAM - FOR DEMONSTRATION (LISTING STARTS ON PAGE 97)


(RELOCATABLE LOCATION 1Ø1)
E
LET I = 1
CTRI = -4
FINL = Ø
CTRP = -WORD COUNT OF PIM DATA
BOUND 1 = 256
BOUND 2 = 512
BOUND 3 = 768
BOUND 4 = 1024
SET OTBS TO BASE ADDRESS OF PIM DATA,
SET AC2 TO OTBS
SET OTCTR = Ø
READ IN THE AMPLITUDE FROM KEYS 8-11
LET THIS = AMP
F
SLOOP
READ IN Y VALUE (CONTENTS OF AC2 + 1).
SWAP HALVES
MASK OFF ALL BUT BITS 8-11.
LET THIS = YAMP.
IS YAMP - AMP < Ø ?
NO
H
YES
G
1A-46,566
G
AC2 = AC2 + 2
CTRP = CTRP + 1
IS CTRP = Ø ?
YES
HALT
(THIS IMPLIES AN ODD WORD COUNT)
NO
CTRP = CTRP + 1
IS CTRP = Ø ?
NO
F
YES
FINL = FINL + 1
I
H
GETY
READ IN Y VALUE (CONTENTS OF AC2 + 1)
J

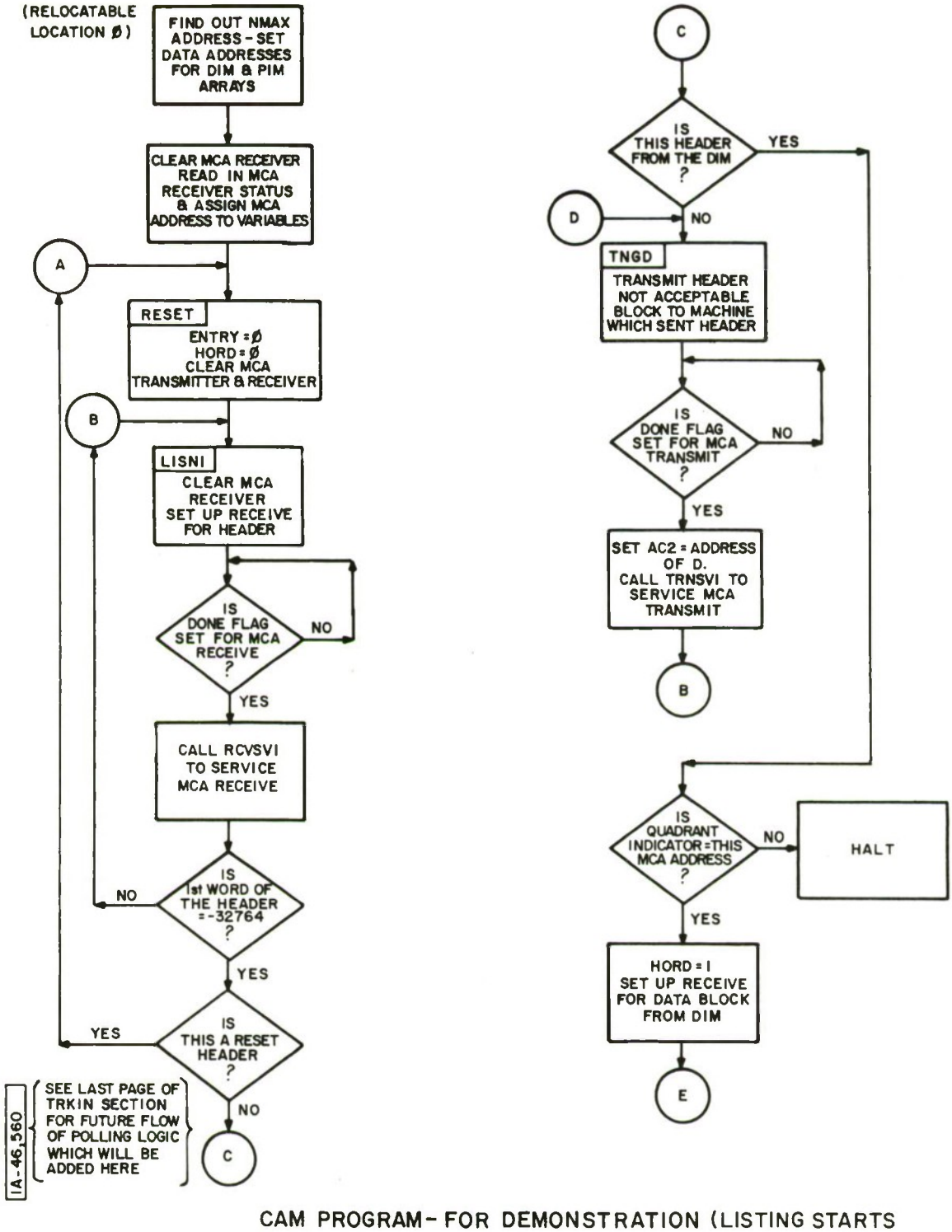
DIM - 2


(RELOCATABLE LOCATION 157)
J
MASK OFF AMPLITUDE BITS 0-3
IS NEW Y < BOUND(i) ?
NO
I
YES
PUT X AND Y VALUES TO OUTPUT ARRAY
OTCTR = OTCTR +2
AC2 = AC2 + 2
CTRP = CTRP + 1
IS CTRP = Ø ?
YES
HALT (THIS IMPLIES AN ODD WORD COUNT)
NO
CTRP = CTRP + 1
IS CTRP = Ø ?
NO
F
YES
FINL = FINL + 1
I
I
SENDC
NXTAD = AC2
CREATE THE HEADER BLOCK TO SEND TO CAM$_i$
K
TRANSMIT HEADER BLOCK TO CAM$_i$
SET UP RECEIVE FOR CAM$_i$ ACKNOWLEDGE
IS DONE FLAG SET FOR MCA TRANSMIT ?
NO
YES
SET AC1 = 1 TO INDICATE HEADER. SET AC2 = ADDRESS OF K.
CALL TRNSV2 TO SERVICE MCA TRANSMIT
IS DONE FLAG SET FOR MCA RECEIVE ?
NO
YES
L
IA-46,567

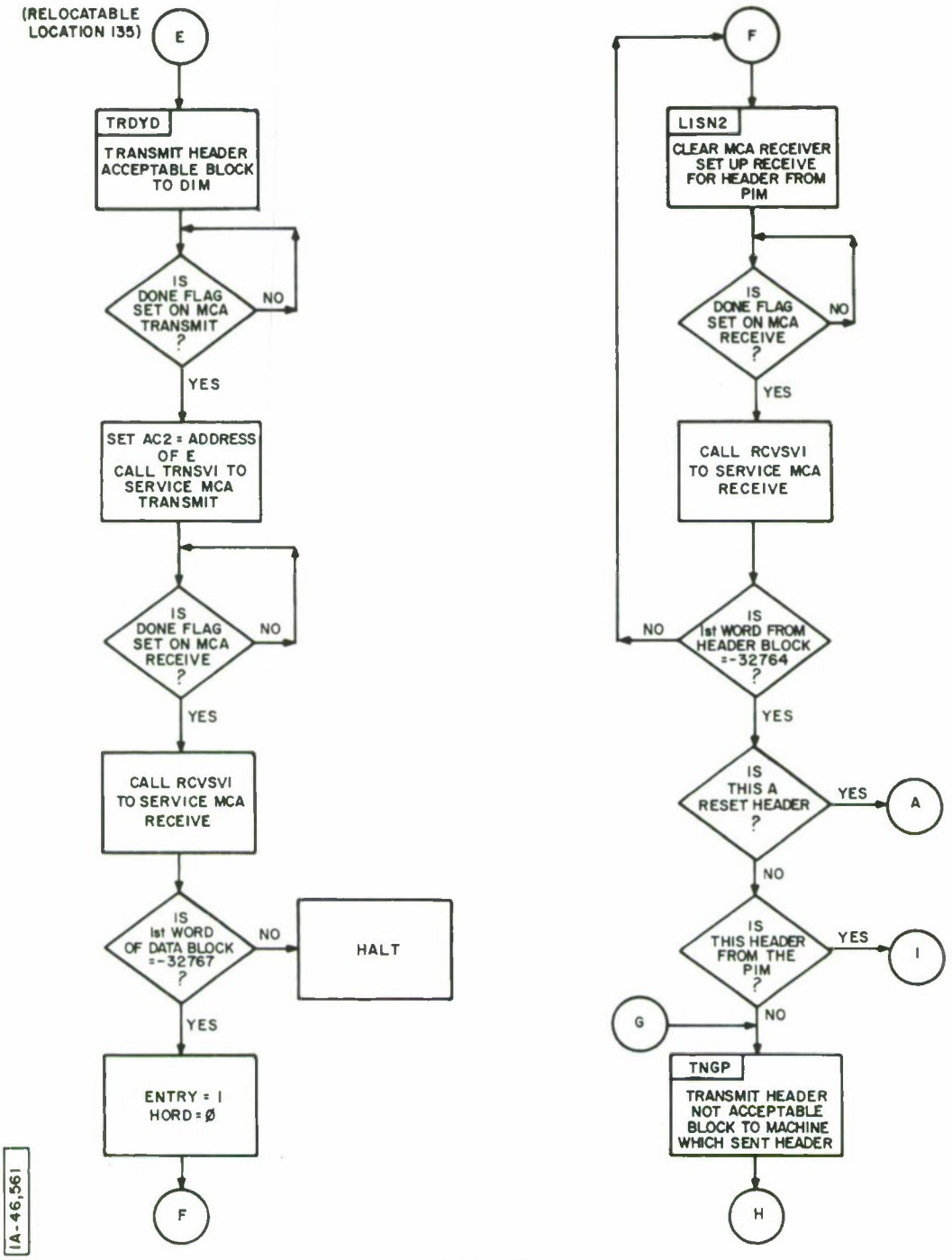
DIM - 3


(RELOCATABLE LOCATION 253)
L
CALL RCVSVI TO SERVICE MCA RECEIVE
IS ACKNOWLEDGE FROM CAMi ?
NO
HALT
YES
IS 1st WORD OF ACKNOWLEDGE =-32765 ?
NO
HALT
YES
WAS HEADER ACCEPTABLE TO CAMi ?
NO
K
M
YES
TRANSMIT OTCTR WORDS FROM ADDRESS OTBAS TO CAMi
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC1 = Ø TO INDICATE OATA
SET AC2 = ADDRESS OF M
N
IA-46,568
N
CALL TRNSV2 TO SERVICE MCA TRANSMIT
I = I + 1
OTBS = NXTAD
OTCTR = Ø
CTRI = CTRI + 1
IS CTRI = Ø ?
YES
A
NO
IS FINL = Ø ?
YES
F
NO
I

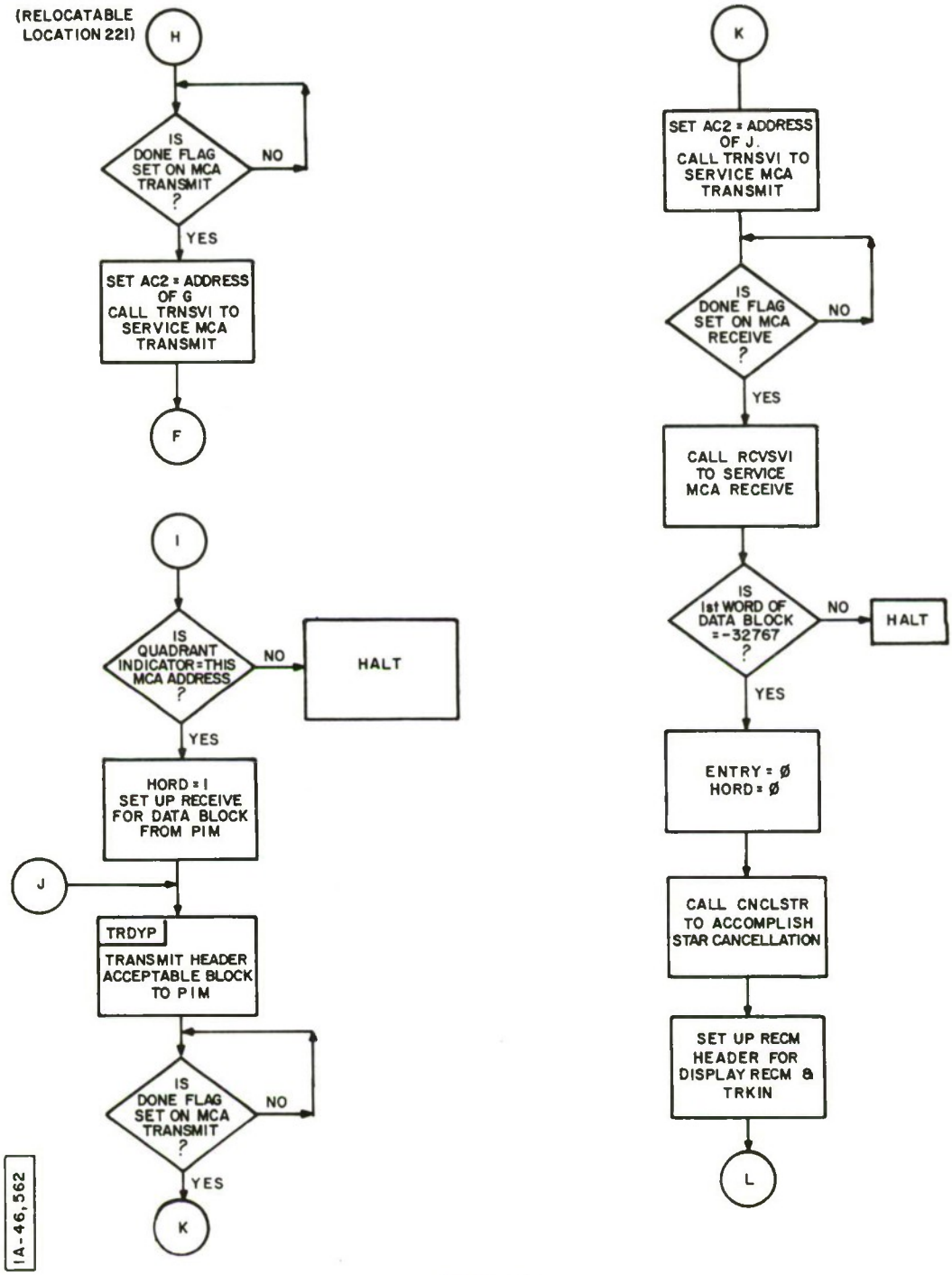
DIM-4



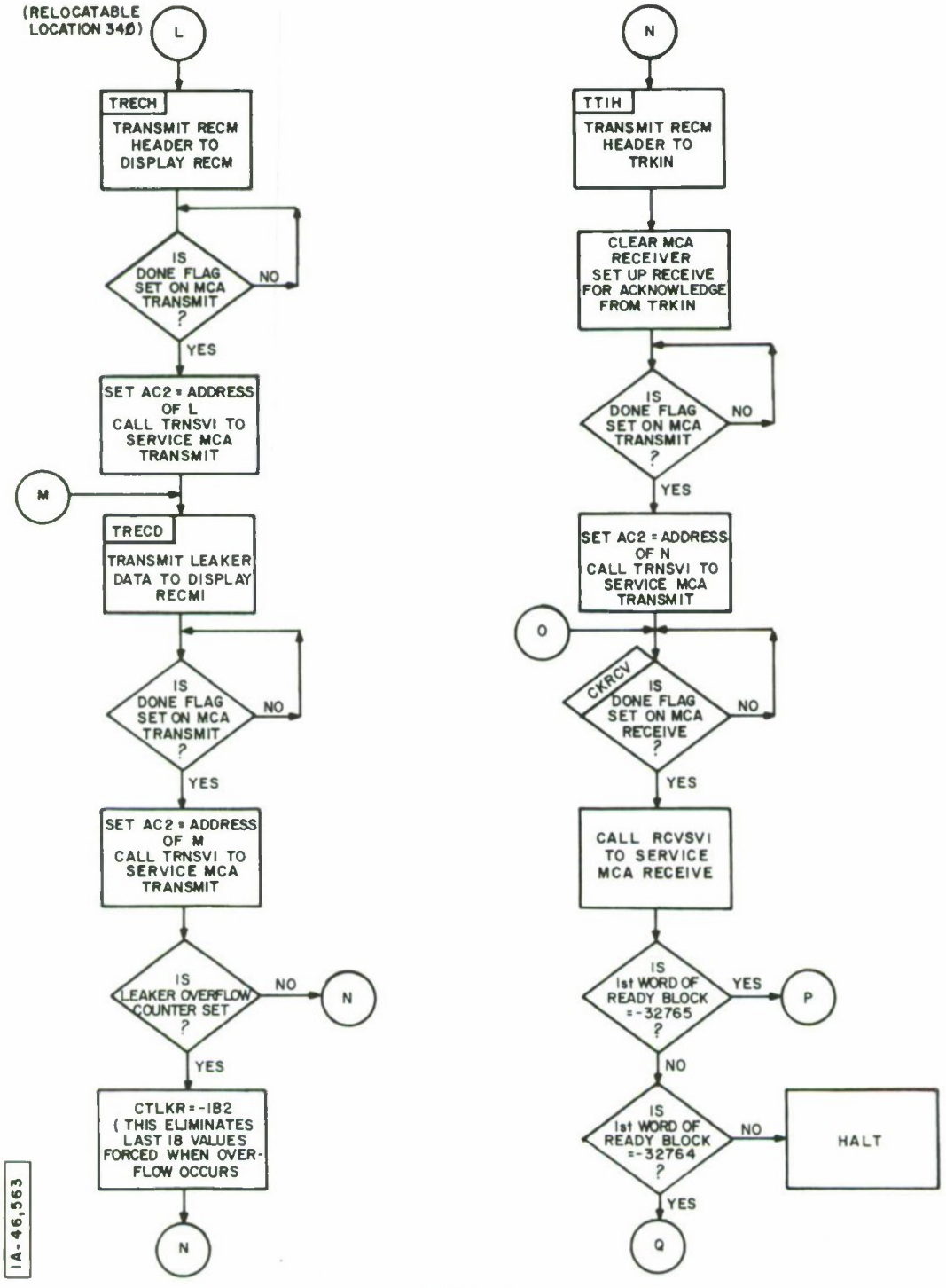
CAM PROGRAM - FOR DEMONSTRATION (LISTING STARTS ON PAGE 104)


(RELOCATABLE LOCATION 135)
E
TRDYD
TRANSMIT HEADER ACCEPTABLE BLOCK TO DIM
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF E
CALL TRNSVI TO SERVICE MCA TRANSMIT
IS DONE FLAG SET ON MCA RECEIVE ?
NO
YES
CALL RCVSVI TO SERVICE MCA RECEIVE
IS 1st WORD OF DATA BLOCK =-32767 ?
NO
HALT
YES
ENTRY = I
HORD = Ø
F
F
LISN2
CLEAR MCA RECEIVER SET UP RECEIVE FOR HEADER FROM PIM
IS DONE FLAG SET ON MCA RECEIVE ?
NO
YES
CALL RCVSVI TO SERVICE MCA RECEIVE
IS 1st WORD FROM HEADER BLOCK =-32764 ?
NO
YES
IS THIS A RESET HEADER ?
YES
A
NO
IS THIS HEADER FROM THE PIM ?
YES
I
NO
G
TNGP
TRANSMIT HEADER NOT ACCEPTABLE BLOCK TO MACHINE WHICH SENT HEADER
H
IA-46,561

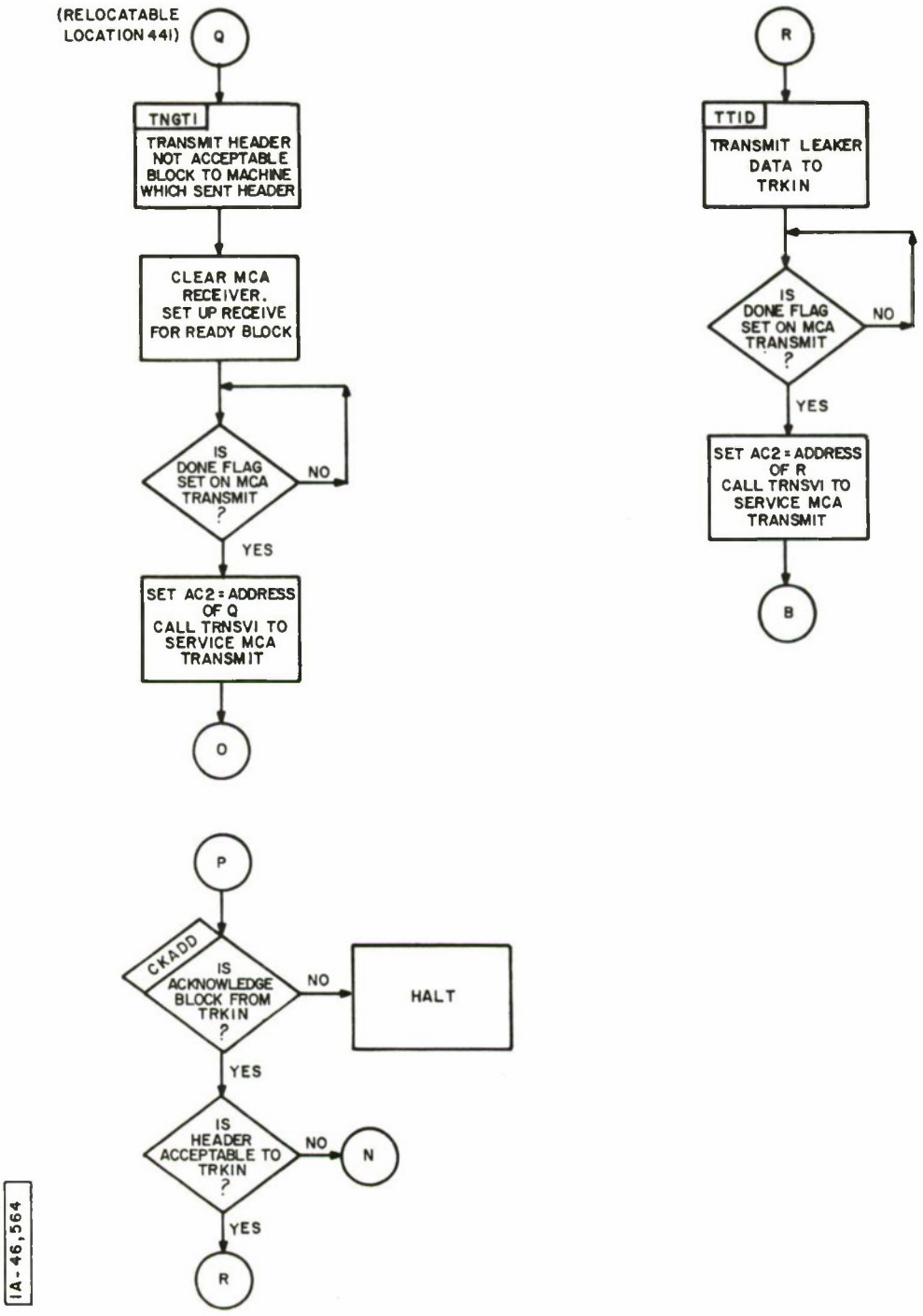
CAM - 2


(RELOCATABLE LOCATION 221)
H
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF G CALL TRNSVI TO SERVICE MCA TRANSMIT
F
I
IS QUADRANT INDICATOR = THIS MCA ADDRESS ?
NO
HALT
YES
HORD = 1 SET UP RECEIVE FOR DATA BLOCK FROM PIM
J
TRDYP
TRANSMIT HEADER ACCEPTABLE BLOCK TO PIM
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
K
IA-46,562
K
SET AC2 = ADDRESS OF J. CALL TRNSVI TO SERVICE MCA TRANSMIT
IS DONE FLAG SET ON MCA RECEIVE ?
NO
YES
CALL RCVSVI TO SERVICE MCA RECEIVE
IS 1st WORD OF DATA BLOCK = -32767 ?
NO
HALT
YES
ENTRY = Ø HORD = Ø
CALL CNCLSTR TO ACCOMPLISH STAR CANCELLATION
SET UP RECM HEADER FOR DISPLAY RECM & TRKIN
L

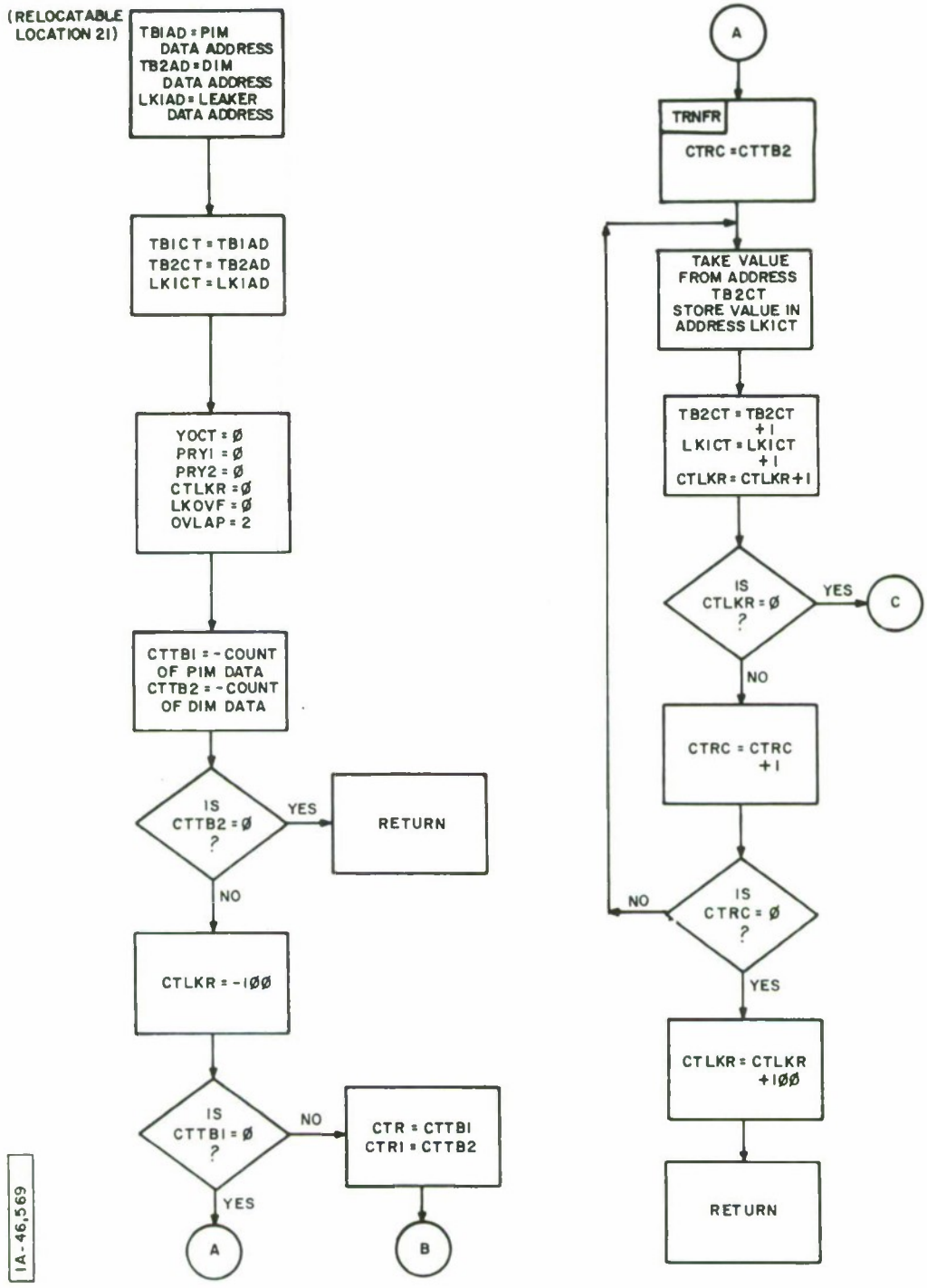
CAM - 3


(RELOCATABLE LOCATION 340)
L
TRECH
TRANSMIT RECM HEADER TO DISPLAY RECM
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF L CALL TRNSVI TO SERVICE MCA TRANSMIT
M
TRECD
TRANSMIT LEAKER DATA TO DISPLAY RECMI
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF M CALL TRNSVI TO SERVICE MCA TRANSMIT
IS LEAKER OVERFLOW COUNTER SET ?
NO
N
YES
CTLKR = -182 (THIS ELIMINATES LAST 18 VALUES FORCED WHEN OVER-FLOW OCCURS
N
IA-46,563
N
TTIH
TRANSMIT RECM HEADER TO TRKIN
CLEAR MCA RECEIVER SET UP RECEIVE FOR ACKNOWLEDGE FROM TRKIN
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF N CALL TRNSVI TO SERVICE MCA TRANSMIT
O
CKRCV
IS DONE FLAG SET ON MCA RECEIVE ?
NO
YES
CALL RCVSVI TO SERVICE MCA RECEIVE
IS 1st WORD OF READY BLOCK = -32765 ?
YES
P
NO
IS 1st WORD OF READY BLOCK = -32764 ?
NO
HALT
YES
Q

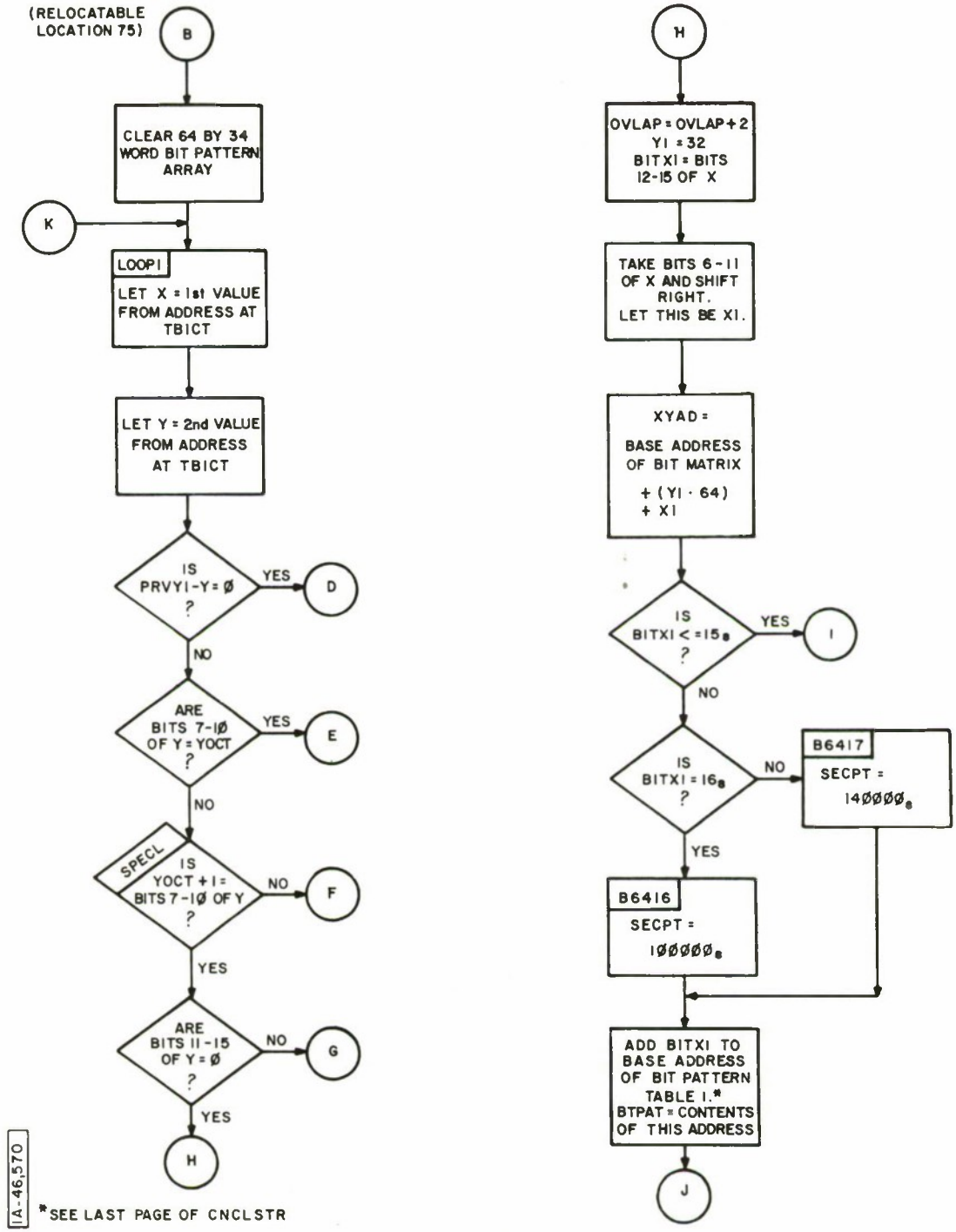
CAM – 4


(RELOCATABLE LOCATION 441)
Q
TNGTI
TRANSMIT HEADER NOT ACCEPTABLE BLOCK TO MACHINE WHICH SENT HEADER
CLEAR MCA RECEIVER, SET UP RECEIVE FOR READY BLOCK
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF Q CALL TRNSVI TO SERVICE MCA TRANSMIT
O
R
TTID
TRANSMIT LEAKER DATA TO TRKIN
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF R CALL TRNSVI TO SERVICE MCA TRANSMIT
B
P
CKADD
IS ACKNOWLEDGE BLOCK FROM TRKIN ?
NO
HALT
YES
IS HEADER ACCEPTABLE TO TRKIN ?
NO
N
YES
R
IA-46,564

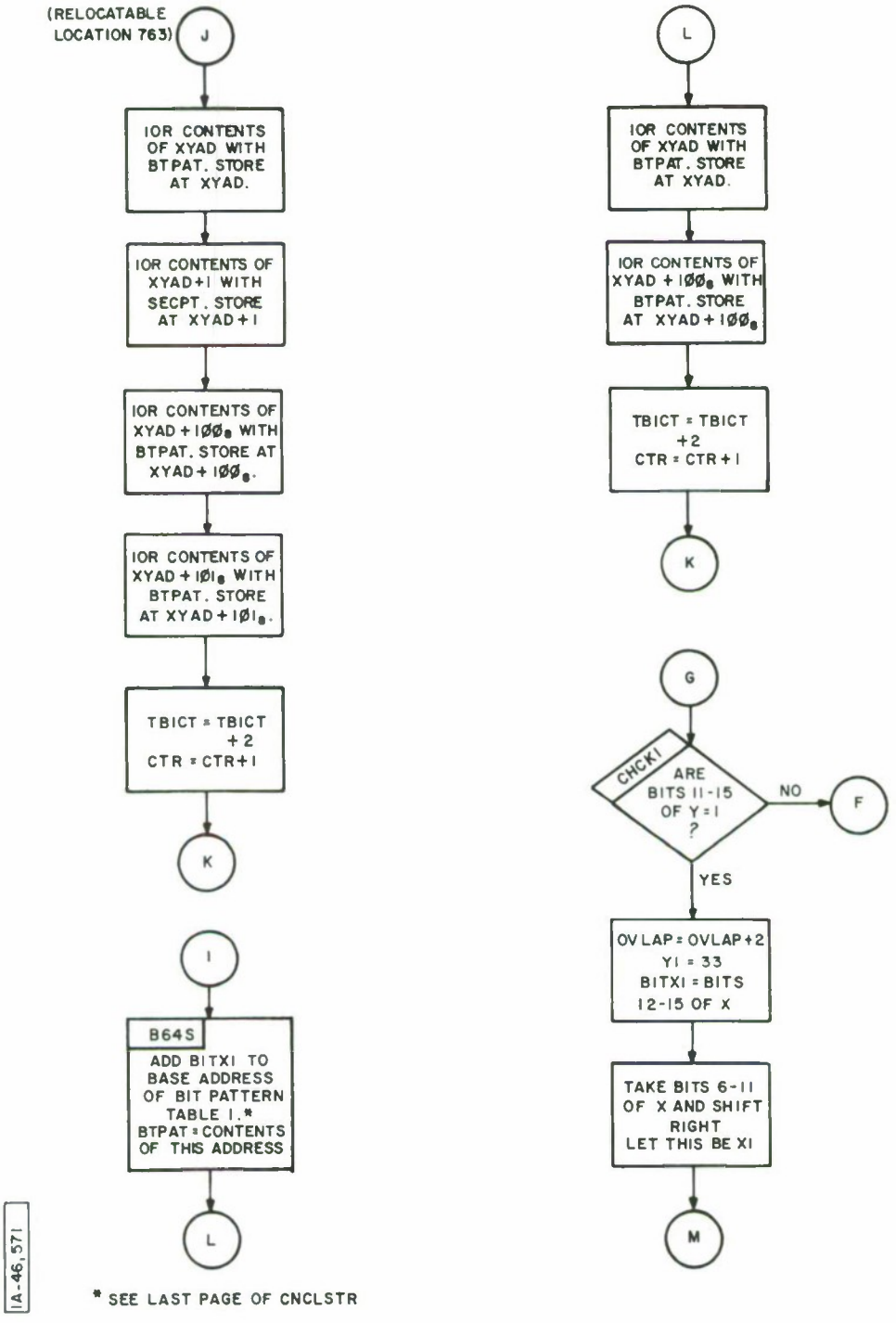
CAM - 5



CNCLSTR - SUBROUTINE OF CAM (LISTING STARTS ON PAGE 113)


(RELOCATABLE LOCATION 75)
B
CLEAR 64 BY 34 WORD BIT PATTERN ARRAY
K
LOOP1
LET X = 1st VALUE FROM ADDRESS AT TBICT
LET Y = 2nd VALUE FROM ADDRESS AT TBICT
IS PRVY1-Y = Ø ?
YES
D
NO
ARE BITS 7-1Ø OF Y = YOCT ?
YES
E
NO
SPECL
IS YOCT + 1 = BITS 7-1Ø OF Y ?
NO
F
YES
ARE BITS 11-15 OF Y = Ø ?
NO
G
YES
H
IA-46,570
*SEE LAST PAGE OF CNCLSTR
H
OVLAP = OVLAP + 2
Y1 = 32
BITX1 = BITS 12-15 OF X
TAKE BITS 6-11 OF X AND SHIFT RIGHT. LET THIS BE X1.
XYAD = BASE ADDRESS OF BIT MATRIX + (Y1 · 64) + X1
IS BITX1 < = 15₈ ?
YES
I
NO
IS BITX1 = 16₈ ?
NO
B6417
SECPT = 14ØØØØ₈
YES
B6416
SECPT = 1ØØØØØ₈
ADD BITX1 TO BASE ADDRESS OF BIT PATTERN TABLE 1.*
BTPAT = CONTENTS OF THIS ADDRESS
J

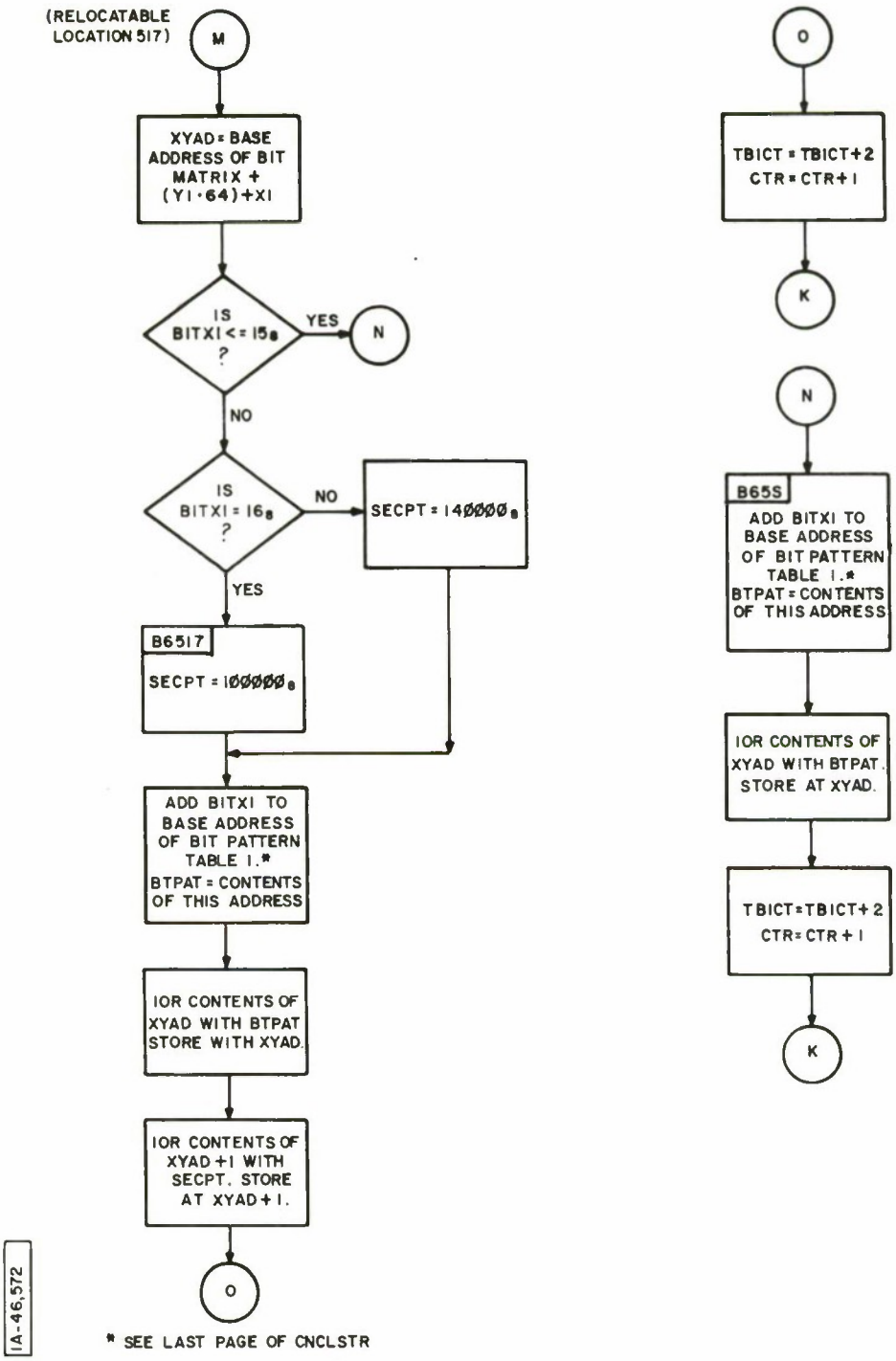
CNCLSTR - 2



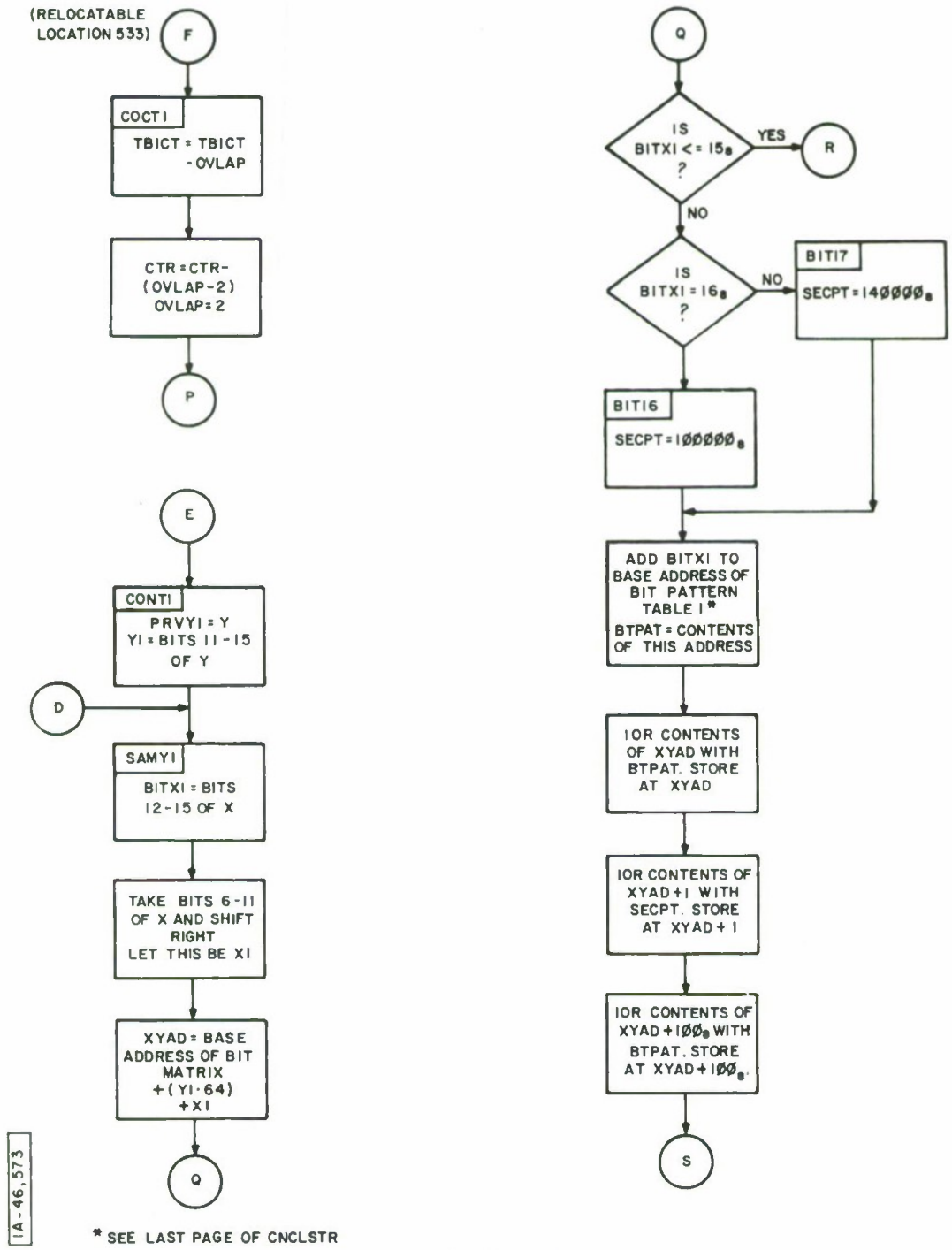
IA-46,571

* SEE LAST PAGE OF CNCLSTR

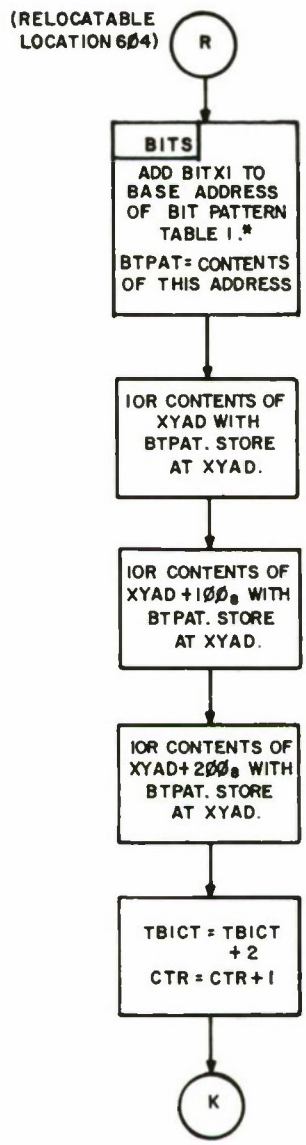
CNCLSTR – 3


(RELOCATABLE LOCATION 517)
M
XYAD = BASE ADDRESS OF BIT MATRIX + (Y1 · 64) + X1
IS BITX1 <= 15₈ ?
YES
N
NO
IS BITX1 = 16₈ ?
NO
SECPT = 140000₈
YES
B6517
SECPT = 100000₈
ADD BITX1 TO BASE ADDRESS OF BIT PATTERN TABLE 1.*
BTPAT = CONTENTS OF THIS ADDRESS
IOR CONTENTS OF XYAD WITH BTPAT STORE WITH XYAD.
IOR CONTENTS OF XYAD + 1 WITH SECPT. STORE AT XYAD + 1.
O
O
TBICT = TBICT + 2
CTR = CTR + 1
K
N
B65S
ADD BITX1 TO BASE ADDRESS OF BIT PATTERN TABLE 1.*
BTPAT = CONTENTS OF THIS ADDRESS
IOR CONTENTS OF XYAD WITH BTPAT. STORE AT XYAD.
TBICT = TBICT + 2
CTR = CTR + 1
K
1A-46,572
* SEE LAST PAGE OF CNCLSTR

CNCLSTR - 4

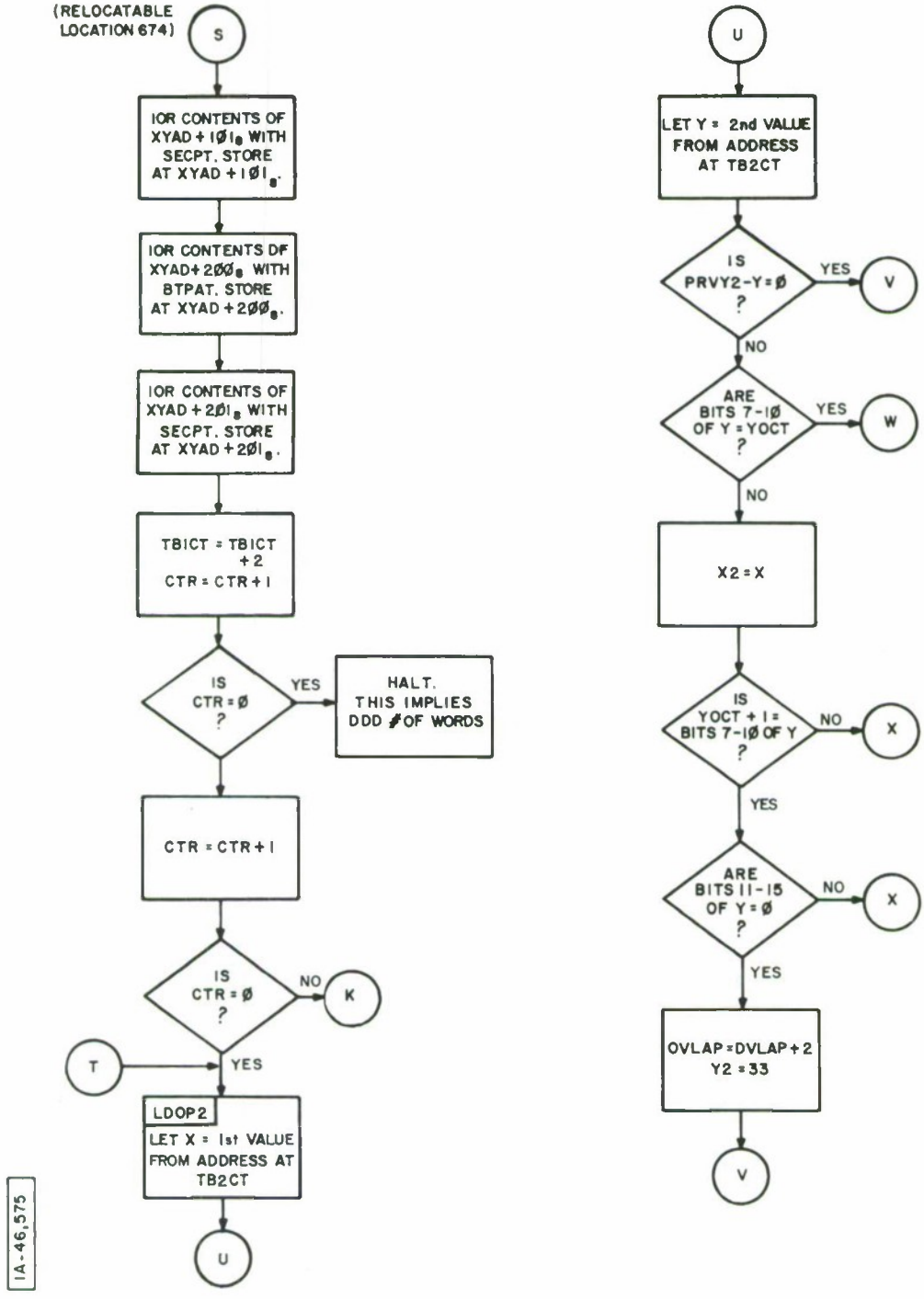

(RELOCATABLE LOCATION 533)
F
COCT1
TBICT = TBICT - OVLAP
CTR = CTR-(OVLAP-2)
OVLAP = 2
P
E
CONT1
PRVY1 = Y
Y1 = BITS 11-15 OF Y
D
SAMY1
BITX1 = BITS 12-15 OF X
TAKE BITS 6-11 OF X AND SHIFT RIGHT
LET THIS BE X1
XYAD = BASE ADDRESS OF BIT MATRIX +(Y1·64) +X1
Q
1A-46,573
* SEE LAST PAGE OF CNCLSTR
Q
IS BITX1 <= 15₈ ?
YES
R
NO
IS BITX1 = 16₈ ?
NO
BIT17
SECPT = 140000₈
BIT16
SECPT = 100000₈
ADD BITX1 TO BASE ADDRESS OF BIT PATTERN TABLE 1*
BTPAT = CONTENTS OF THIS ADDRESS
IOR CONTENTS OF XYAD WITH BTPAT. STORE AT XYAD
IOR CONTENTS OF XYAD+1 WITH SECPT. STORE AT XYAD+1
IOR CONTENTS OF XYAD+100₈ WITH BTPAT. STORE AT XYAD+100₈
S

CNCLSTR - 5

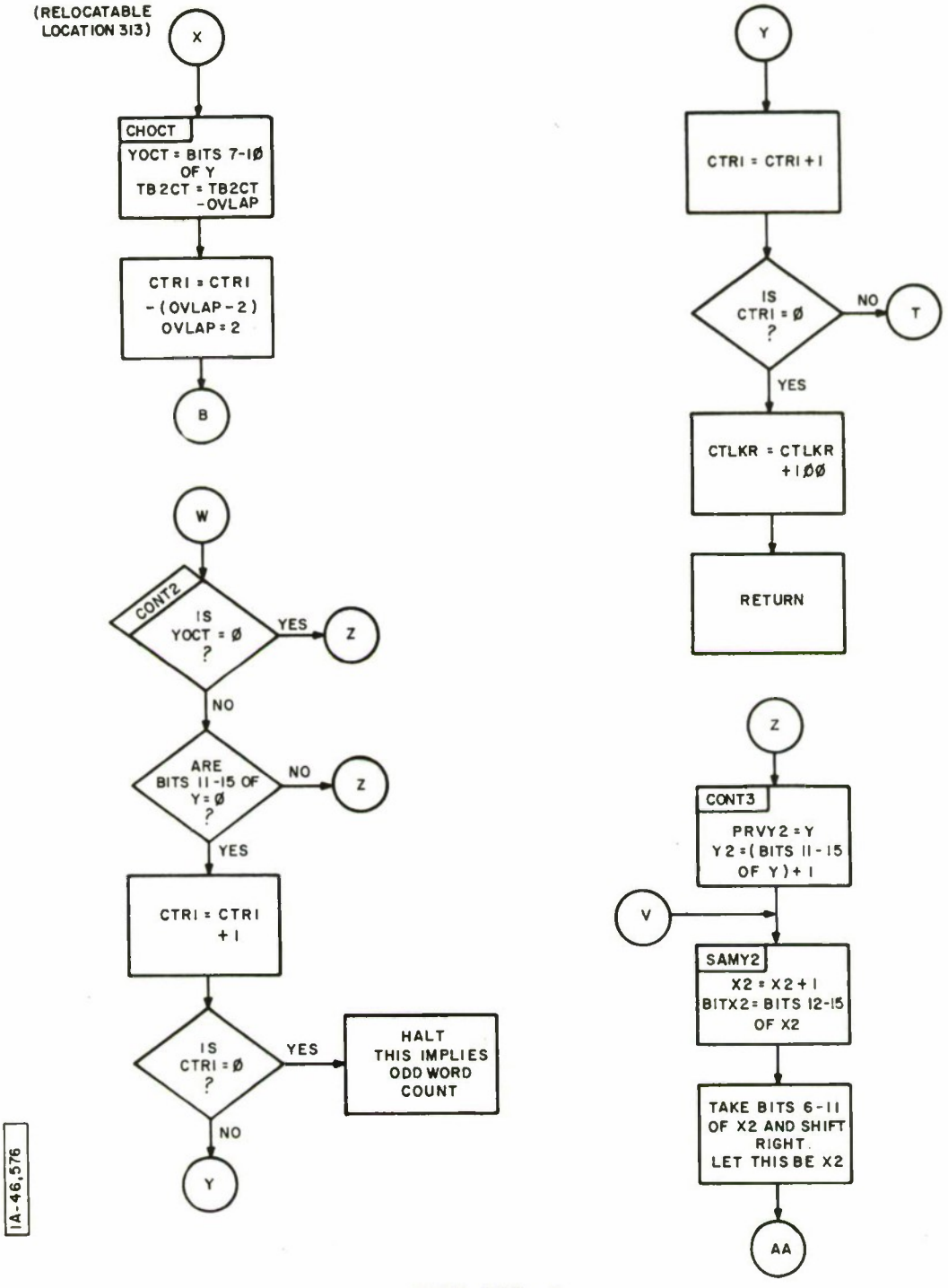


*SEE LAST PAGE OF CNCLSTR

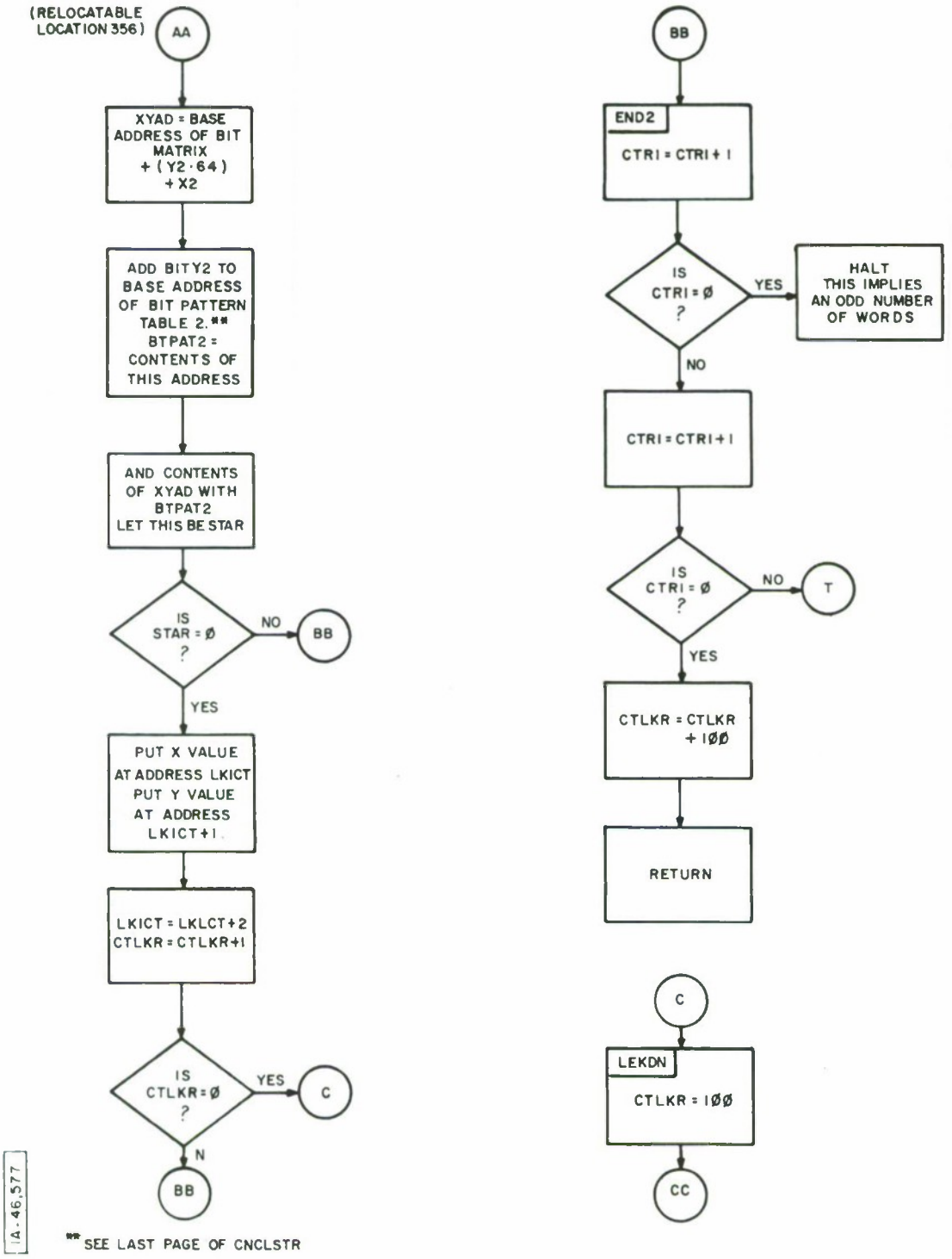
IA-46,574


(RELOCATABLE LOCATION 674)
S
IOR CONTENTS OF XYAD + 1Ø1₈ WITH SECPT. STORE AT XYAD + 1Ø1₈.
IOR CONTENTS DF XYAD+ 2ØØ₈ WITH BTPAT. STORE AT XYAD + 2ØØ₈.
IOR CONTENTS OF XYAD + 2Ø1₈ WITH SECPT. STORE AT XYAD + 2Ø1₈.
TBICT = TBICT +2
CTR = CTR + 1
IS CTR = Ø ?
YES
HALT. THIS IMPLIES DDD # OF WORDS
CTR = CTR + 1
IS CTR = Ø ?
NO
K
T
YES
LDOP2
LET X = 1st VALUE FROM ADDRESS AT TB2CT
U
1A-46,575
U
LET Y = 2nd VALUE FROM ADDRESS AT TB2CT
IS PRVY2-Y = Ø ?
YES
V
NO
ARE BITS 7-1Ø OF Y = YOCT ?
YES
W
NO
X2 = X
IS YOCT + 1 = BITS 7-1Ø OF Y ?
NO
X
YES
ARE BITS 11-15 OF Y = Ø ?
NO
X
YES
OVLAP = DVLAP + 2
Y2 = 33
V

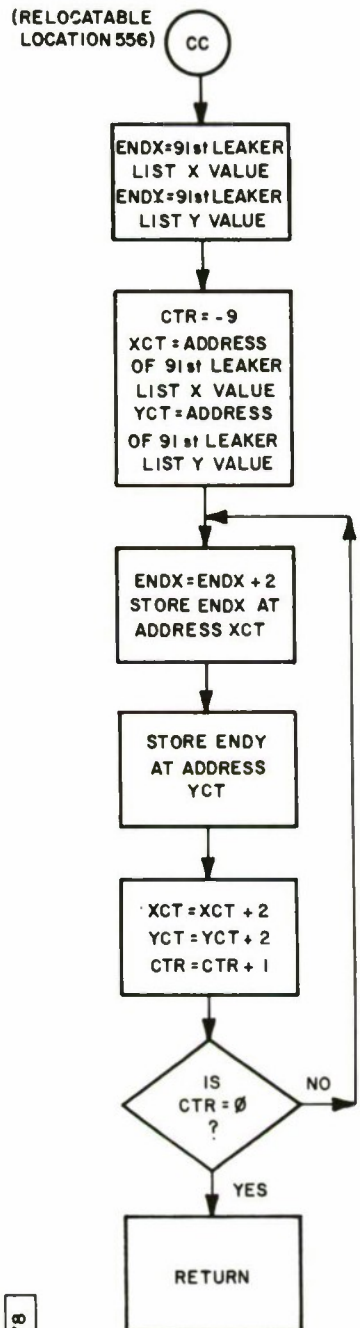
CNCLSTR - 7


(RELOCATABLE LOCATION 313)
X
CHOCT
YOCT = BITS 7-10 OF Y
TB2CT = TB2CT -OVLAP
CTRI = CTRI - (OVLAP-2)
OVLAP = 2
B
W
CONT2
IS YOCT = Ø ?
YES
Z
NO
ARE BITS 11-15 OF Y = Ø ?
NO
Z
YES
CTRI = CTRI + 1
IS CTRI = Ø ?
YES
HALT THIS IMPLIES ODD WORD COUNT
NO
Y
Y
CTRI = CTRI + 1
IS CTRI = Ø ?
NO
T
YES
CTLKR = CTLKR + 1ØØ
RETURN
Z
CONT3
PRVY2 = Y
Y2 = (BITS 11-15 OF Y) + 1
V
SAMY2
X2 = X2 + 1
BITX2 = BITS 12-15 OF X2
TAKE BITS 6-11 OF X2 AND SHIFT RIGHT. LET THIS BE X2
AA
1A-46,576

CNCLSTR - 8


(RELOCATABLE LOCATION 356)
AA
XYAD = BASE ADDRESS OF BIT MATRIX + (Y2·64) + X2
ADD BITY2 TO BASE ADDRESS OF BIT PATTERN TABLE 2.** BTPAT2 = CONTENTS OF THIS ADDRESS
AND CONTENTS OF XYAD WITH BTPAT2 LET THIS BE STAR
IS STAR = Ø ?
NO
BB
YES
PUT X VALUE AT ADDRESS LKICT PUT Y VALUE AT ADDRESS LKICT+1
LKICT = LKLCT+2
CTLKR = CTLKR+1
IS CTLKR = Ø ?
YES
C
N
BB
IA-46,577
** SEE LAST PAGE OF CNCLSTR
BB
END2
CTRI = CTRI + 1
IS CTRI = Ø ?
YES
HALT THIS IMPLIES AN ODD NUMBER OF WORDS
NO
CTRI = CTRI + 1
IS CTRI = Ø ?
NO
T
YES
CTLKR = CTLKR + 1ØØ
RETURN
C
LEKDN
CTLKR = 1ØØ
CC

CNCLSTR - 9


(RELOCATABLE LOCATION 556)
CC
ENDX=91st LEAKER LIST X VALUE
ENDX=91st LEAKER LIST Y VALUE
CTR=-9
XCT=ADDRESS OF 91st LEAKER LIST X VALUE
YCT=ADDRESS OF 91st LEAKER LIST Y VALUE
ENDX=ENDX+2
STORE ENDX AT ADDRESS XCT
STORE ENDY AT ADDRESS YCT
XCT=XCT+2
YCT=YCT+2
CTR=CTR+1
IS CTR=Ø ?
NO
YES
RETURN
IA-46,578

CNCLSTR-10

(RELOCATABLE LOCATION 5323)

*

BIT PATTERN TABLE 1
160000
070000
034000
016000
007000
003400
001600
000700
000340
000160
000070
000034
000016
000007
000003
000001
(ALL VALUES ARE OCTAL)

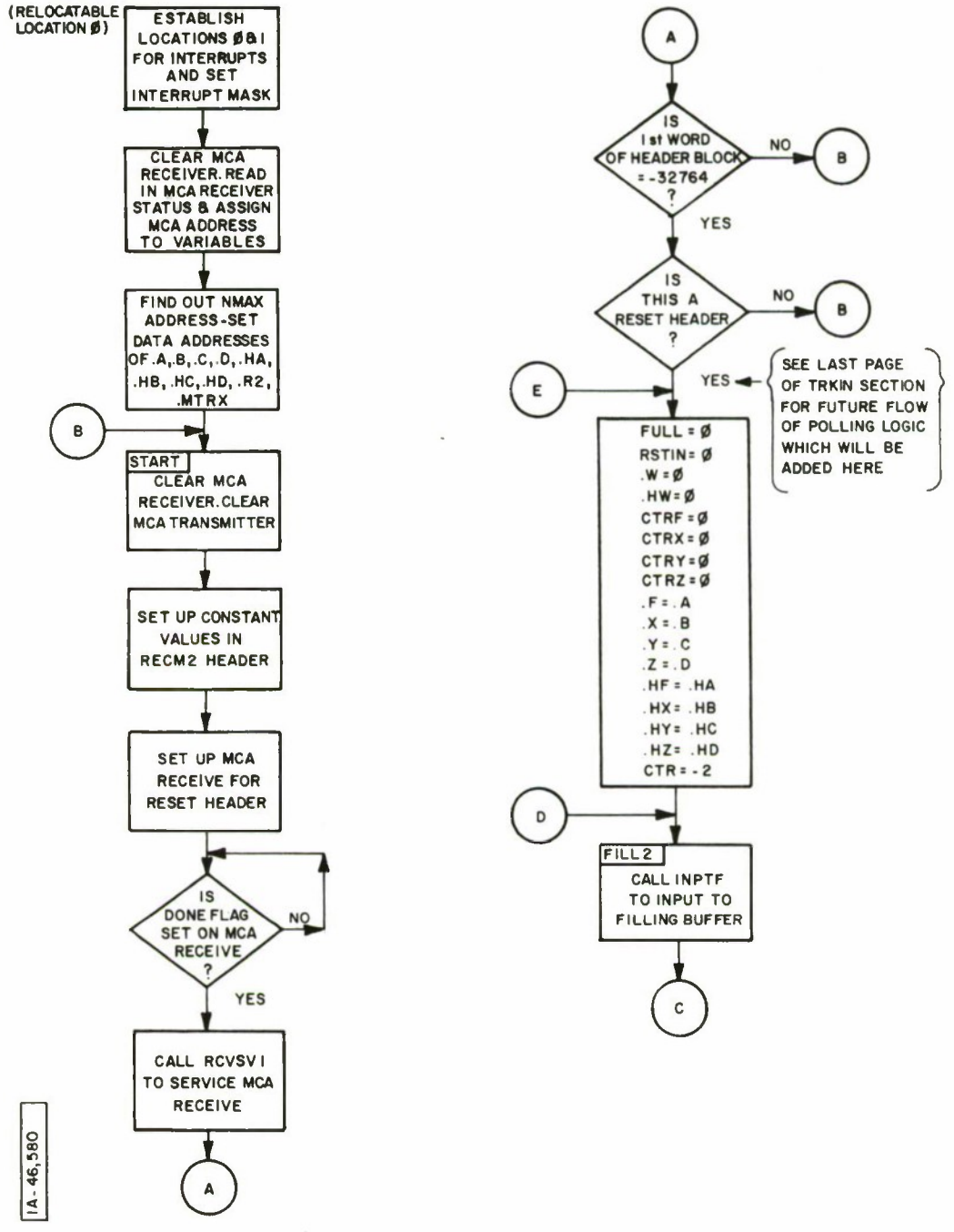
(RELOCATABLE LOCATION 5423)

**

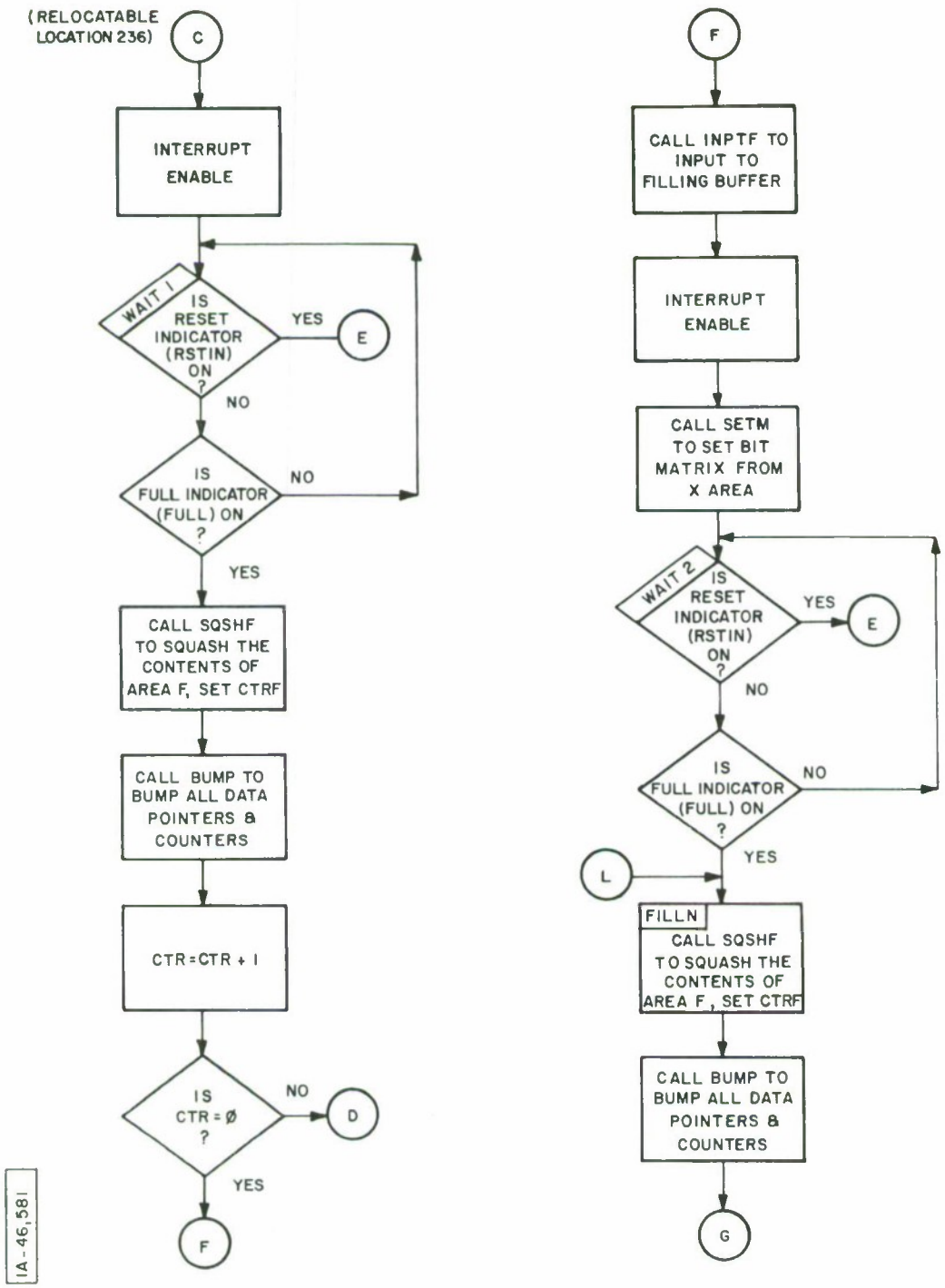
BIT PATTERN TABLE 2
100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001
(ALL VALUES ARE OCTAL)

IA-46,579

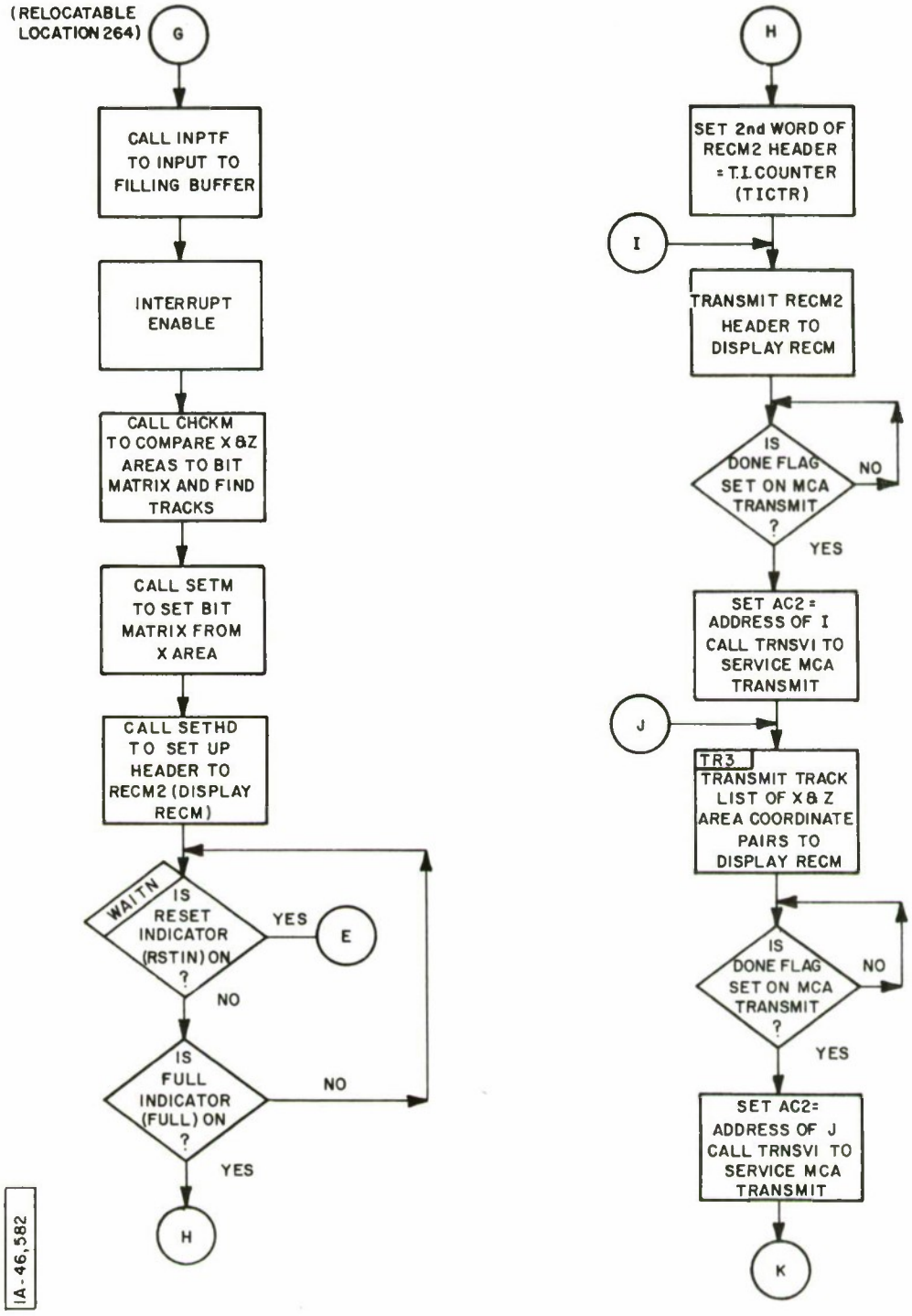
CNCLSTR-11



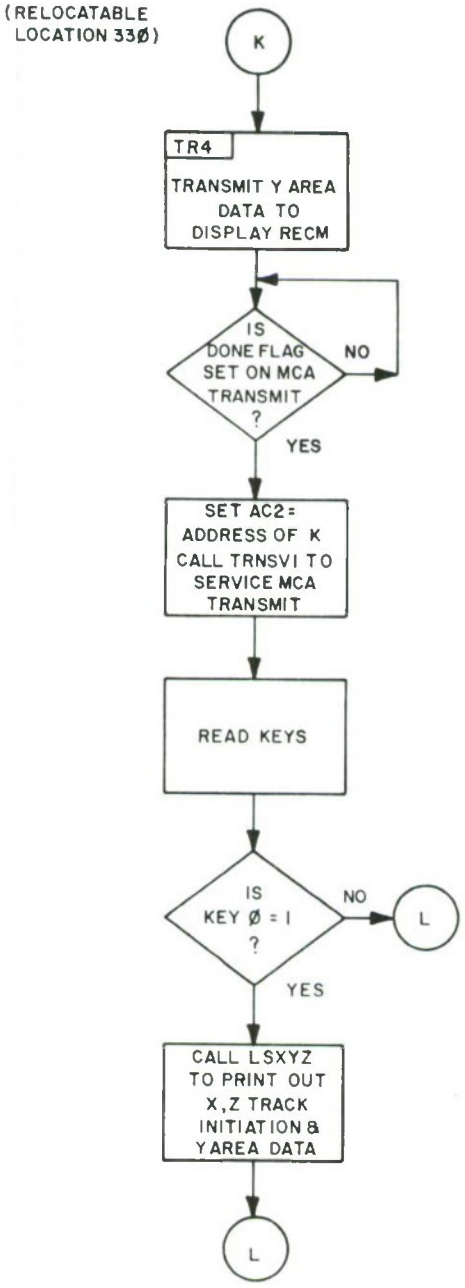
TRKIN - FOR DEMONSTRATION (LISTING STARTS ON PAGE 129)


(RELOCATABLE LOCATION 236)
C
INTERRUPT ENABLE
WAIT 1
IS RESET INDICATOR (RSTIN) ON ?
YES
E
NO
IS FULL INDICATOR (FULL) ON ?
NO
YES
CALL SQSHF TO SQUASH THE CONTENTS OF AREA F, SET CTRF
CALL BUMP TO BUMP ALL DATA POINTERS & COUNTERS
CTR=CTR + 1
IS CTR=Ø ?
NO
D
YES
F
IA-46,581
F
CALL INPTF TO INPUT TO FILLING BUFFER
INTERRUPT ENABLE
CALL SETM TO SET BIT MATRIX FROM X AREA
WAIT 2
IS RESET INDICATOR (RSTIN) ON ?
YES
E
NO
IS FULL INDICATOR (FULL) ON ?
NO
YES
L
FILLN
CALL SQSHF TO SQUASH THE CONTENTS OF AREA F, SET CTRF
CALL BUMP TO BUMP ALL DATA POINTERS & COUNTERS
G

TRKIN - 2

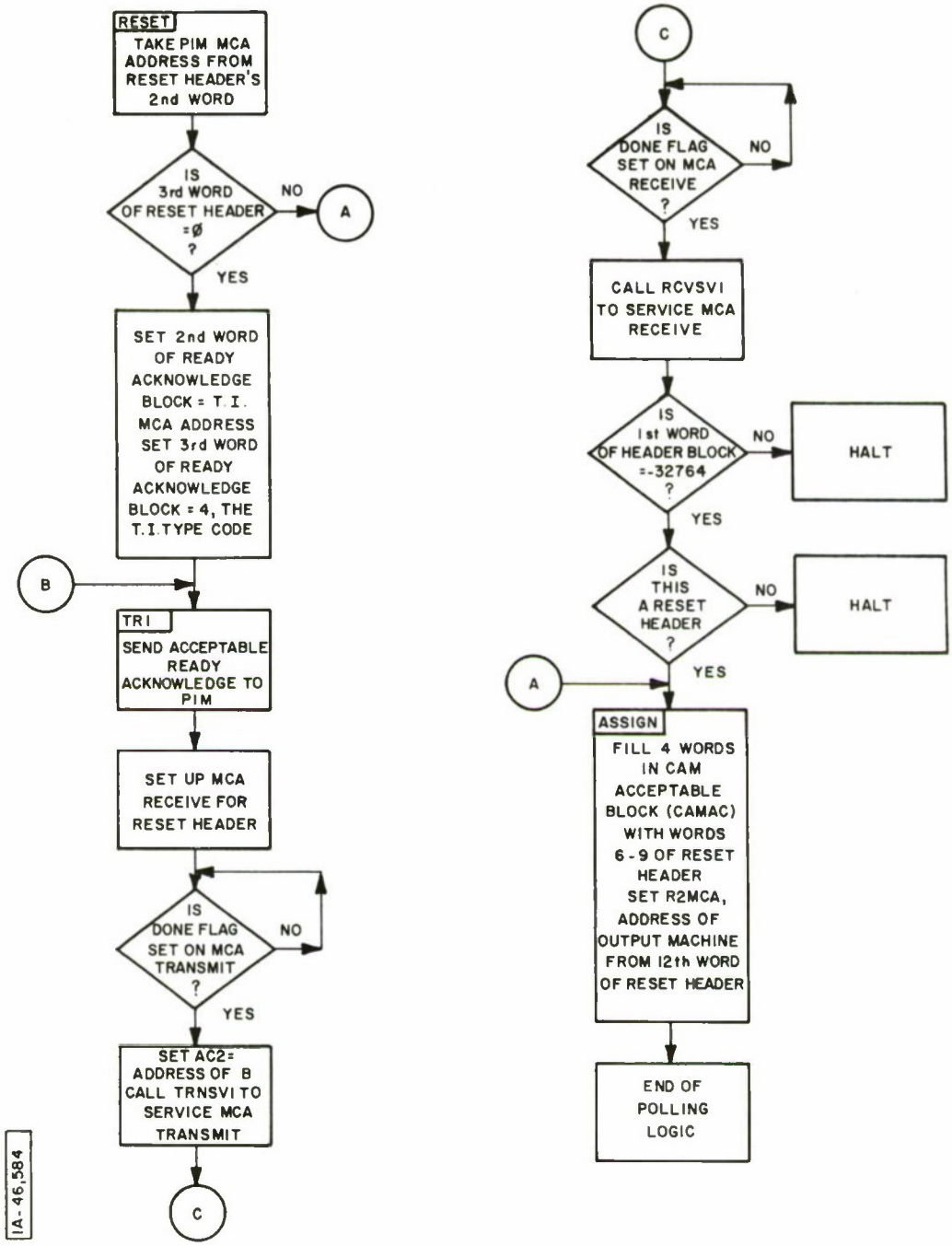

(RELOCATABLE LOCATION 264)
G
CALL INPTF TO INPUT TO FILLING BUFFER
INTERRUPT ENABLE
CALL CHCKM TO COMPARE X & Z AREAS TO BIT MATRIX AND FIND TRACKS
CALL SETM TO SET BIT MATRIX FROM X AREA
CALL SETHD TO SET UP HEADER TO RECM2 (DISPLAY RECM)
WAITN
IS RESET INDICATOR (RSTIN) ON ?
YES
E
NO
IS FULL INDICATOR (FULL) ON ?
NO
YES
H
IA-46,582
H
SET 2nd WORD OF RECM2 HEADER = T.I. COUNTER (TICTR)
I
TRANSMIT RECM2 HEADER TO DISPLAY RECM
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2 = ADDRESS OF I CALL TRNSVI TO SERVICE MCA TRANSMIT
J
TR3
TRANSMIT TRACK LIST OF X & Z AREA COORDINATE PAIRS TO DISPLAY RECM
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
SET AC2= ADDRESS OF J CALL TRNSVI TO SERVICE MCA TRANSMIT
K

TRKIN - 3



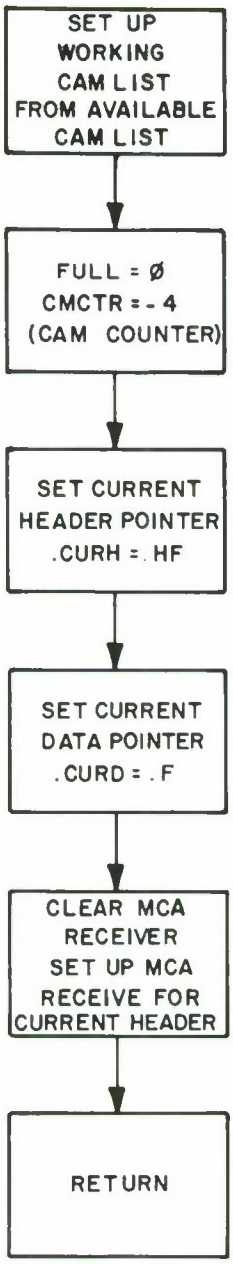
IA-46,583

TRKIN-4



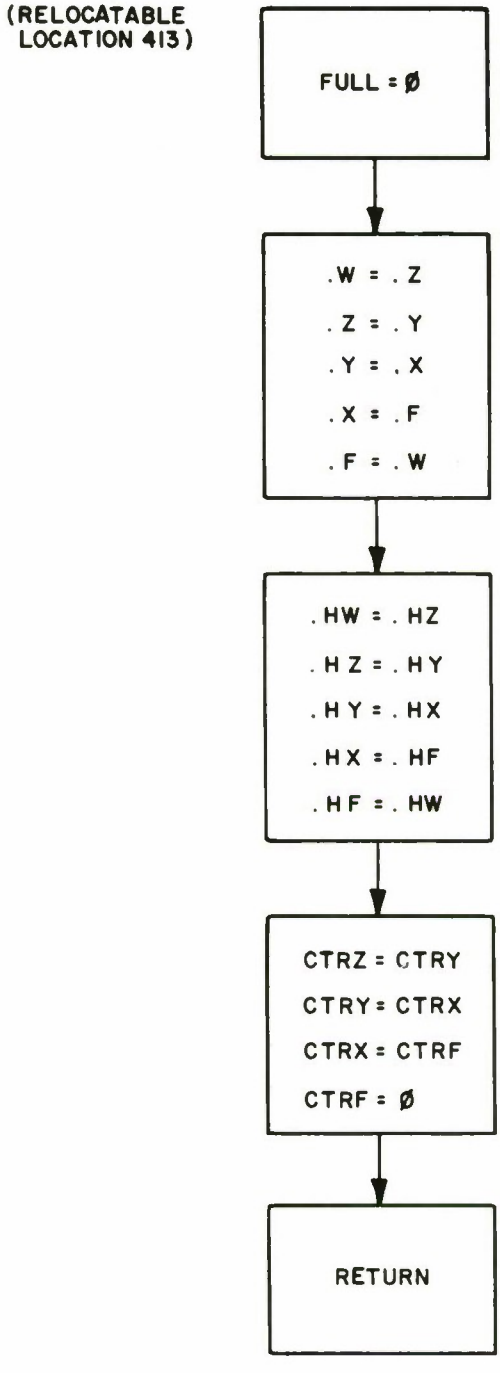
TRKIN-5 POLLING LOGIC - TO BE ADDED WHEN THE MCA TRANSMITTER TIME OUT FUNCTION IS RESTORED

(RELOCATABLE LOCATION 361)



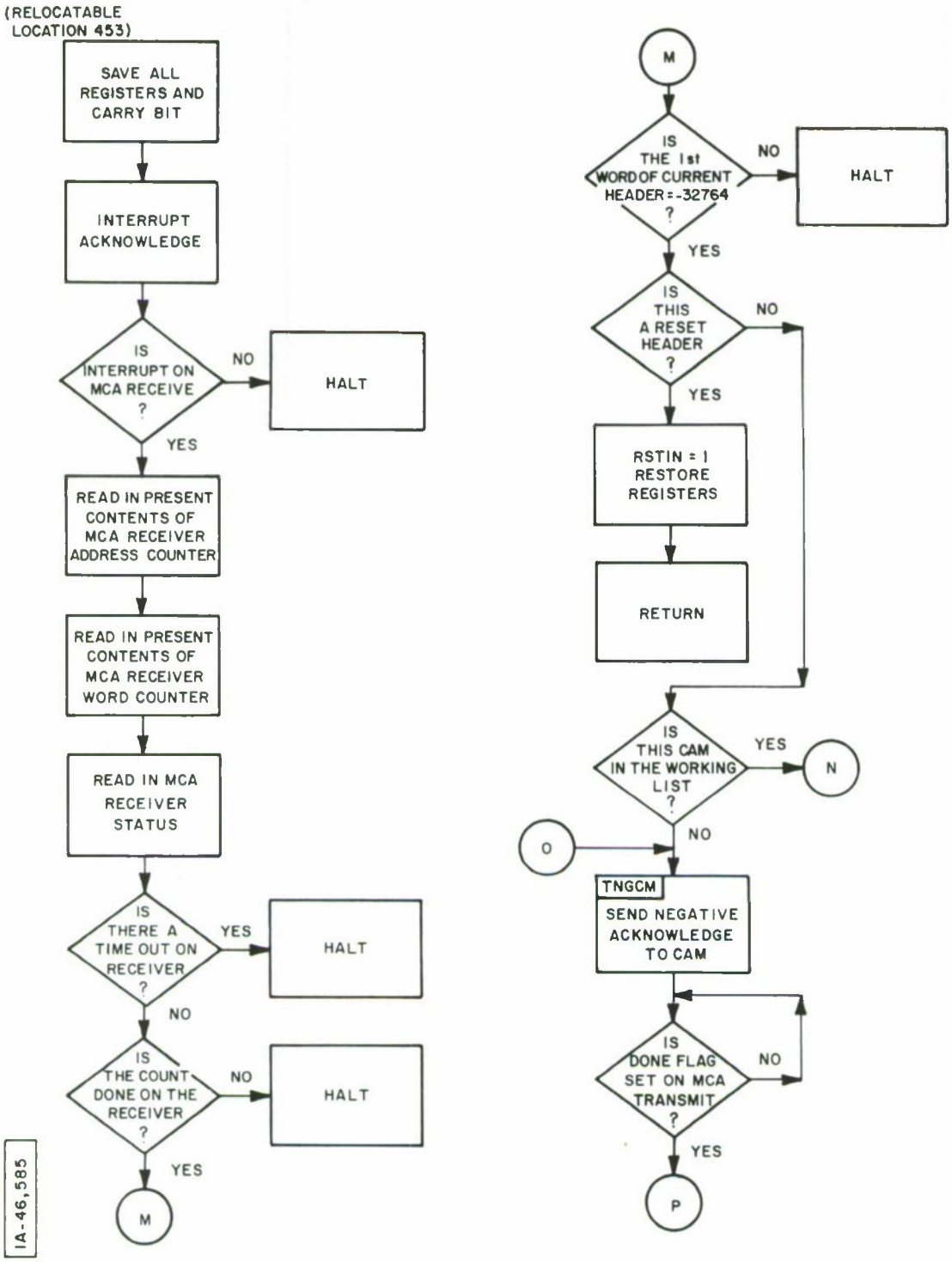
IA-46,598

INPTF - SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 136)

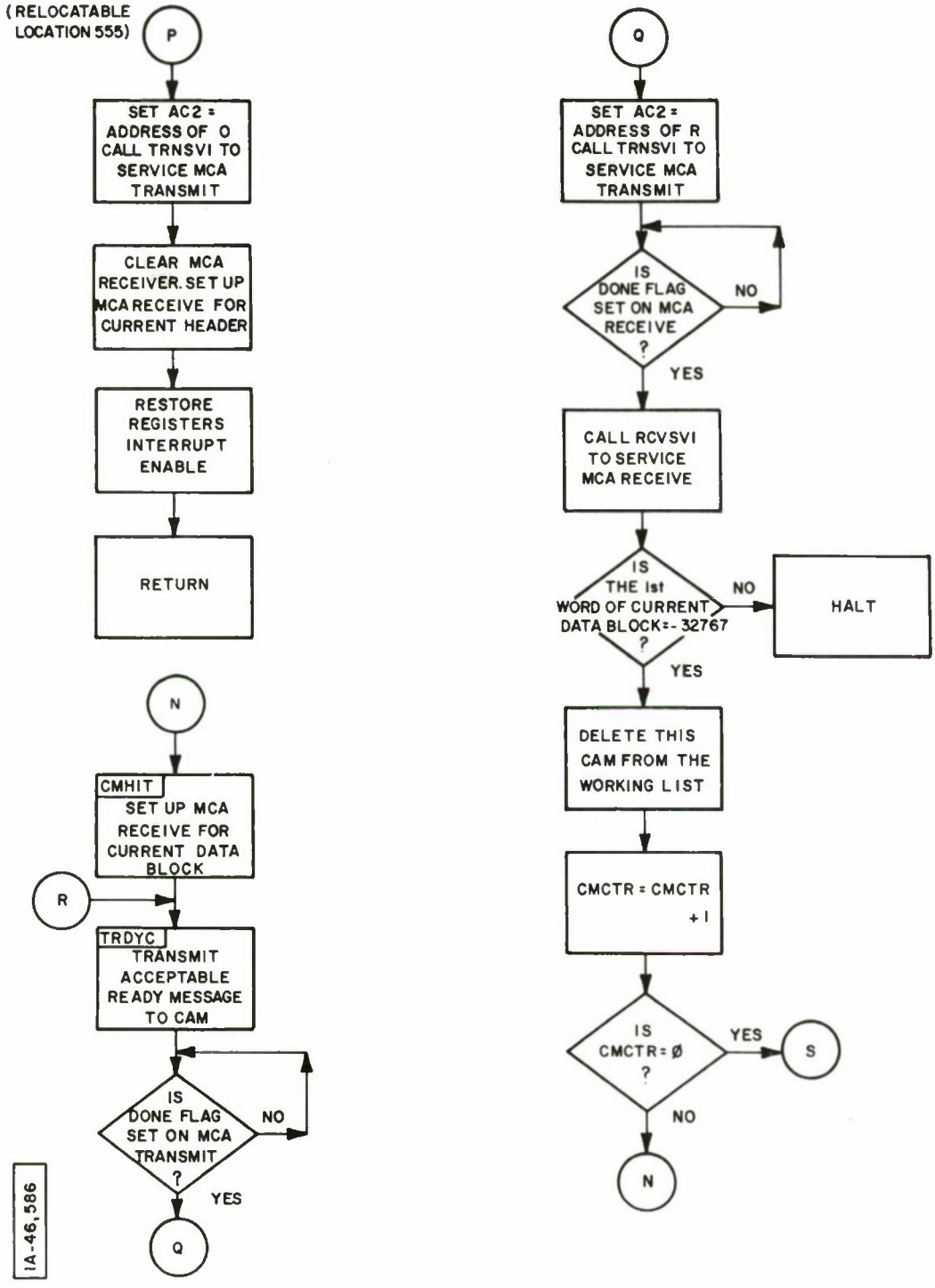


IA-46,597

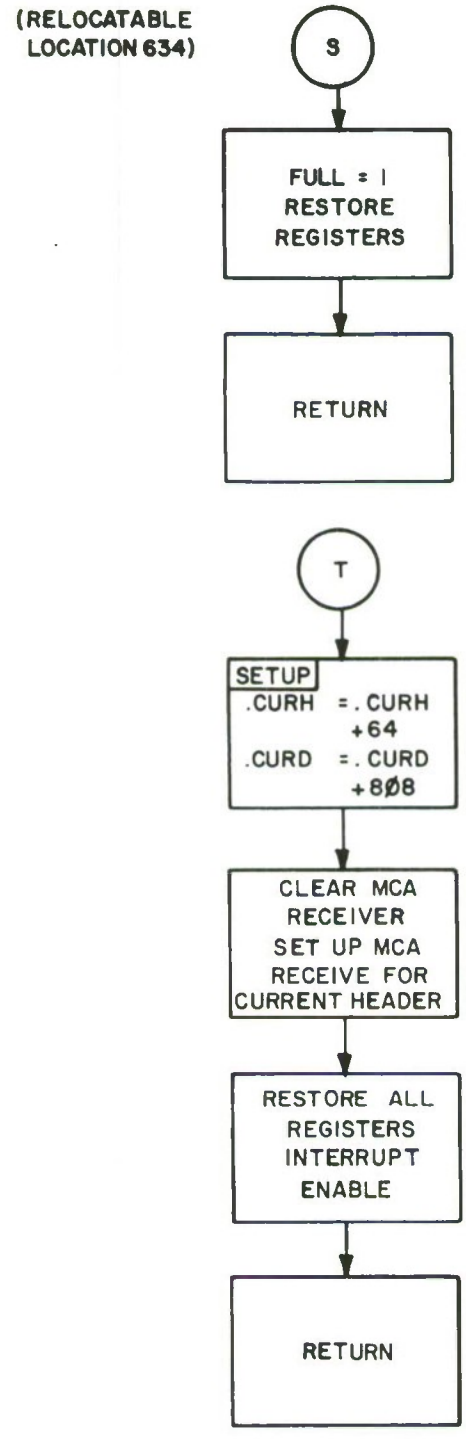
BUMP - SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 137)



INTSV - INTERRUPT SERVICE ROUTINE OF TRKIN (LISTING STARTS ON PAGE 138)

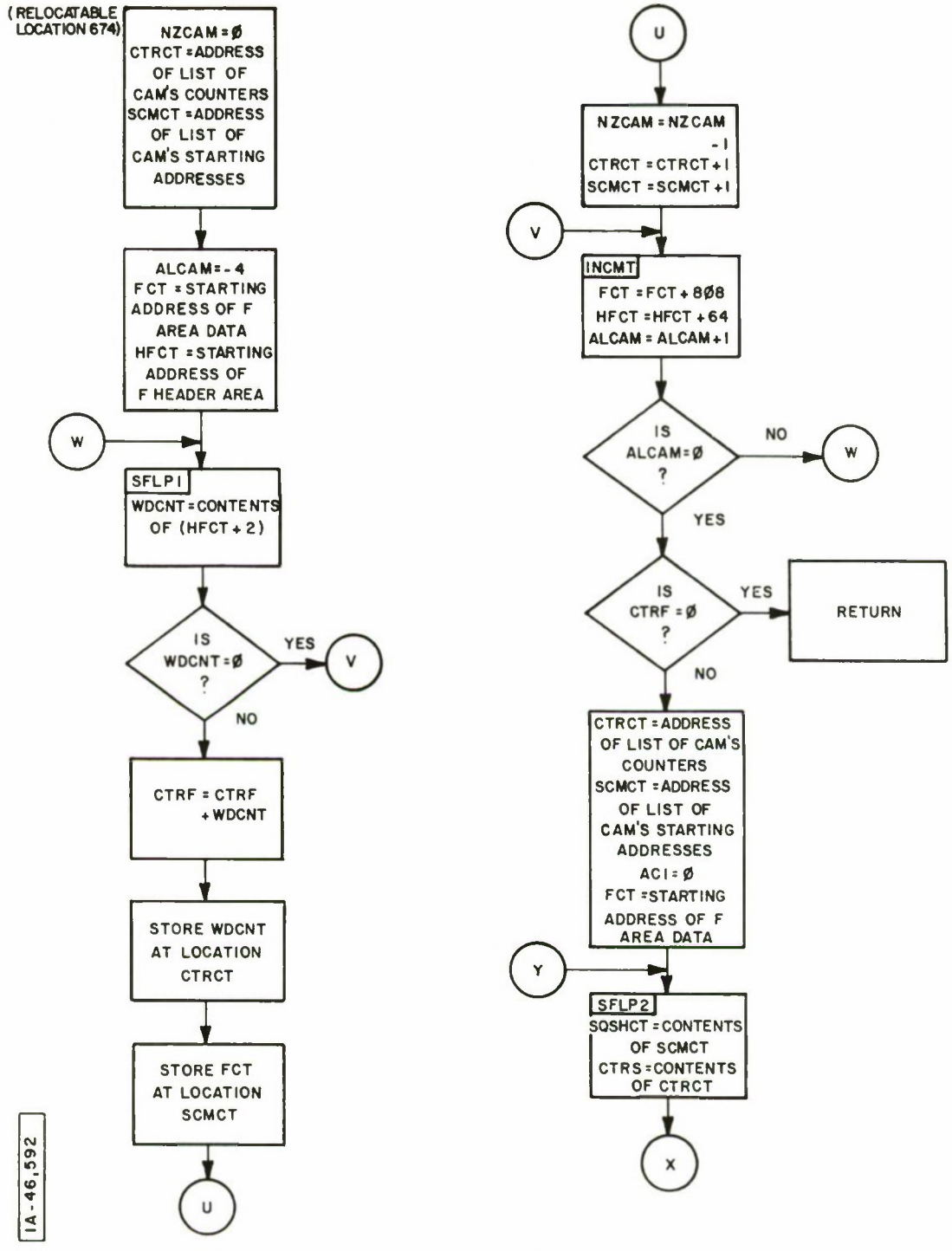

(RELOCATABLE LOCATION 555)
P
SET AC2 = ADDRESS OF O CALL TRNSVI TO SERVICE MCA TRANSMIT
CLEAR MCA RECEIVER. SET UP MCA RECEIVE FOR CURRENT HEADER
RESTORE REGISTERS INTERRUPT ENABLE
RETURN
N
CMHIT
SET UP MCA RECEIVE FOR CURRENT DATA BLOCK
R
TRDYC
TRANSMIT ACCEPTABLE READY MESSAGE TO CAM
IS DONE FLAG SET ON MCA TRANSMIT ?
NO
YES
Q
IA-46,586
Q
SET AC2 = ADDRESS OF R CALL TRNSVI TO SERVICE MCA TRANSMIT
IS DONE FLAG SET ON MCA RECEIVE ?
NO
YES
CALL RCVSVI TO SERVICE MCA RECEIVE
IS THE 1st WORD OF CURRENT DATA BLOCK=-32767 ?
NO
HALT
YES
DELETE THIS CAM FROM THE WORKING LIST
CMCTR = CMCTR + 1
IS CMCTR=Ø ?
YES
S
NO
N

INTSV-2

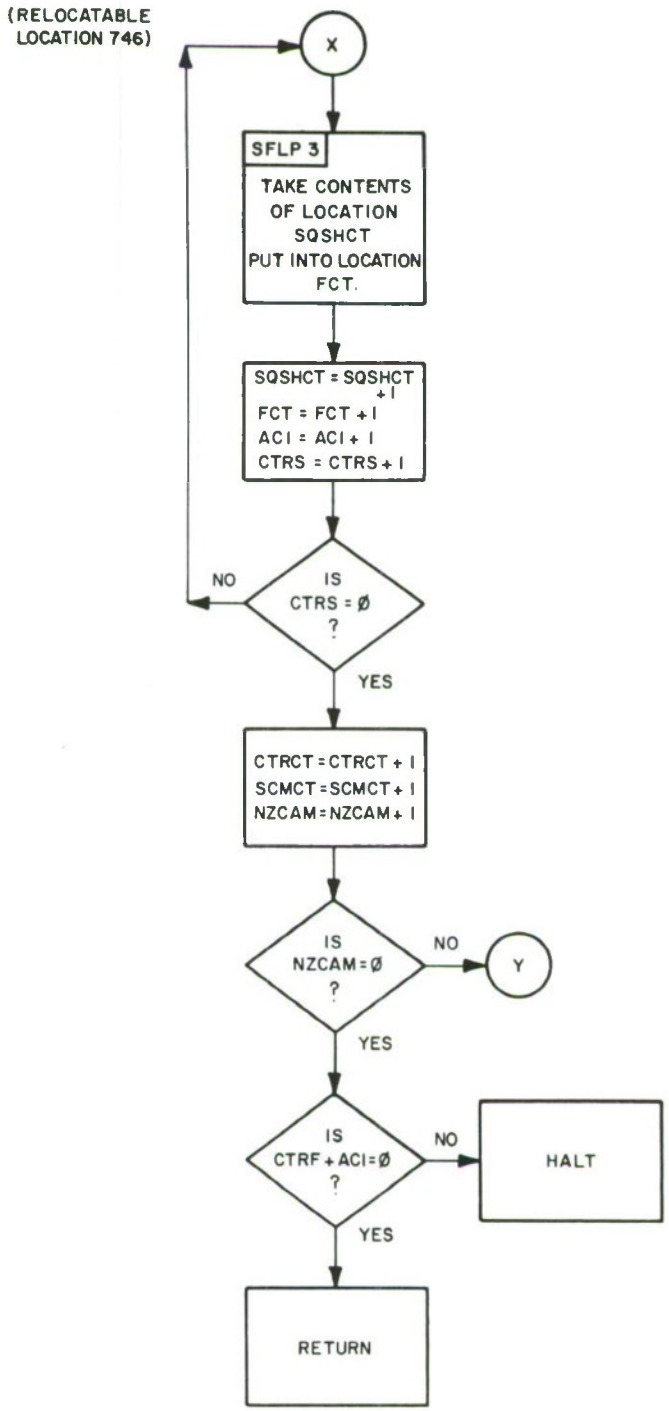


IA-46,387

INTSV - 3

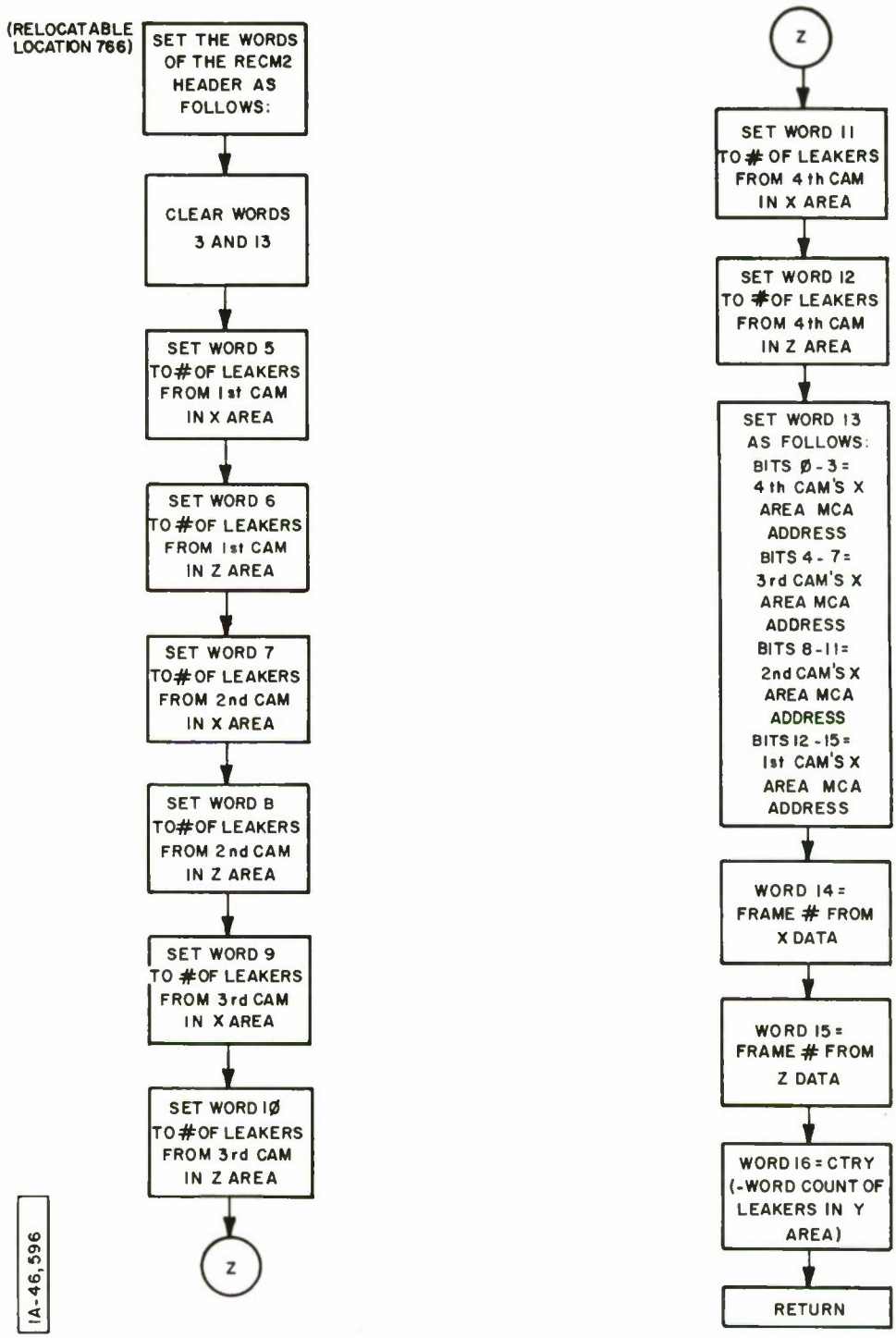


SQSHF - SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 140)

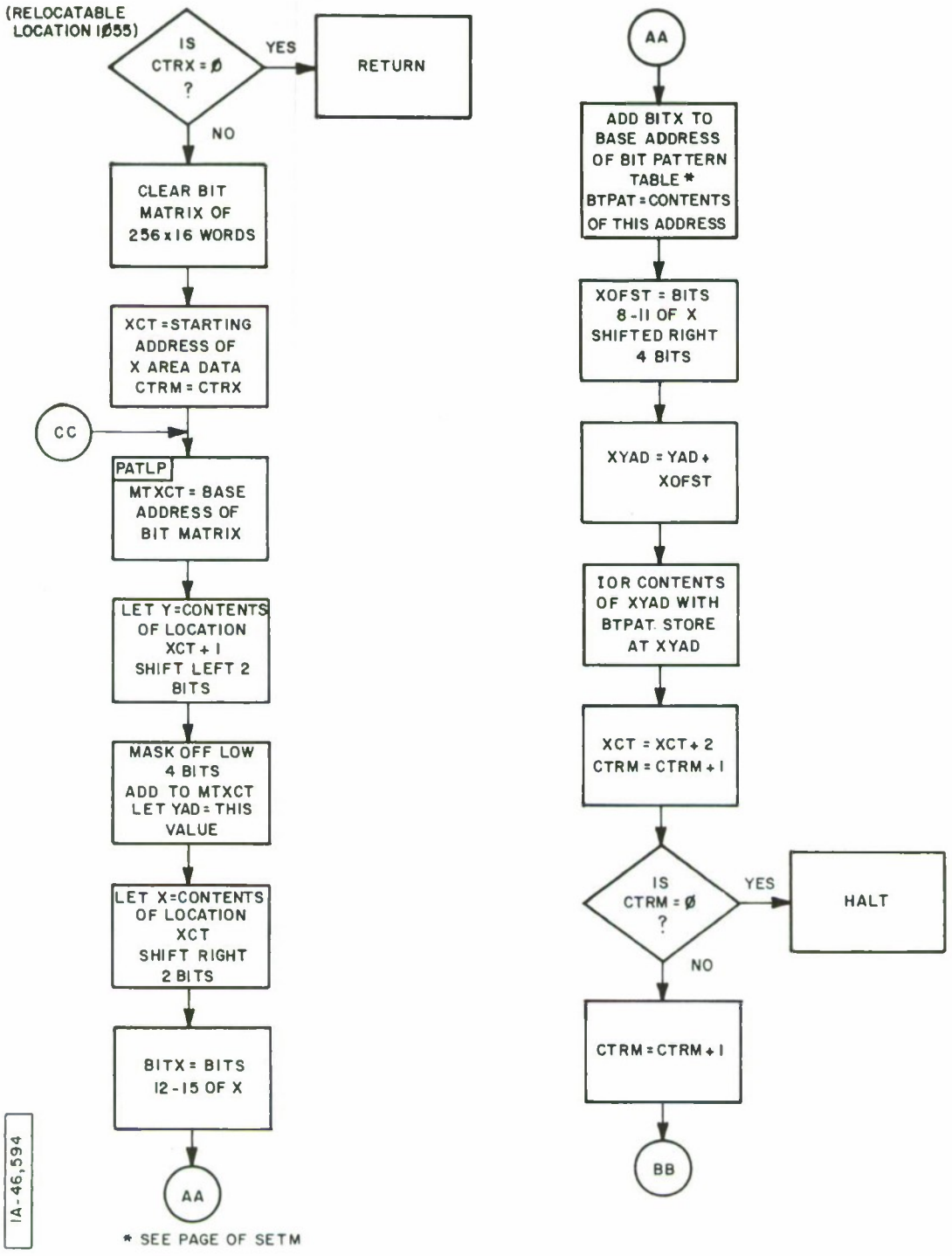


IA-46,593

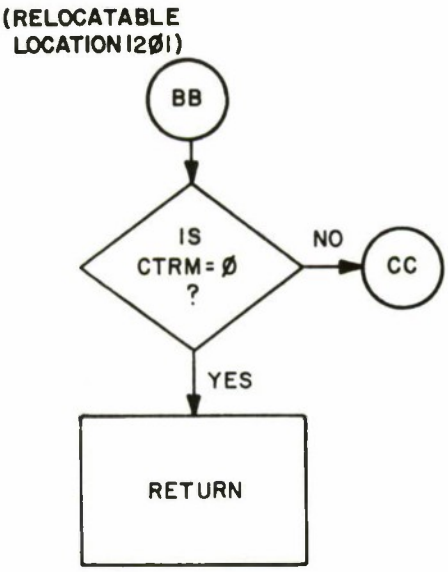
SQSHF-2



SETHD - SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 141)



SETM - SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 142)

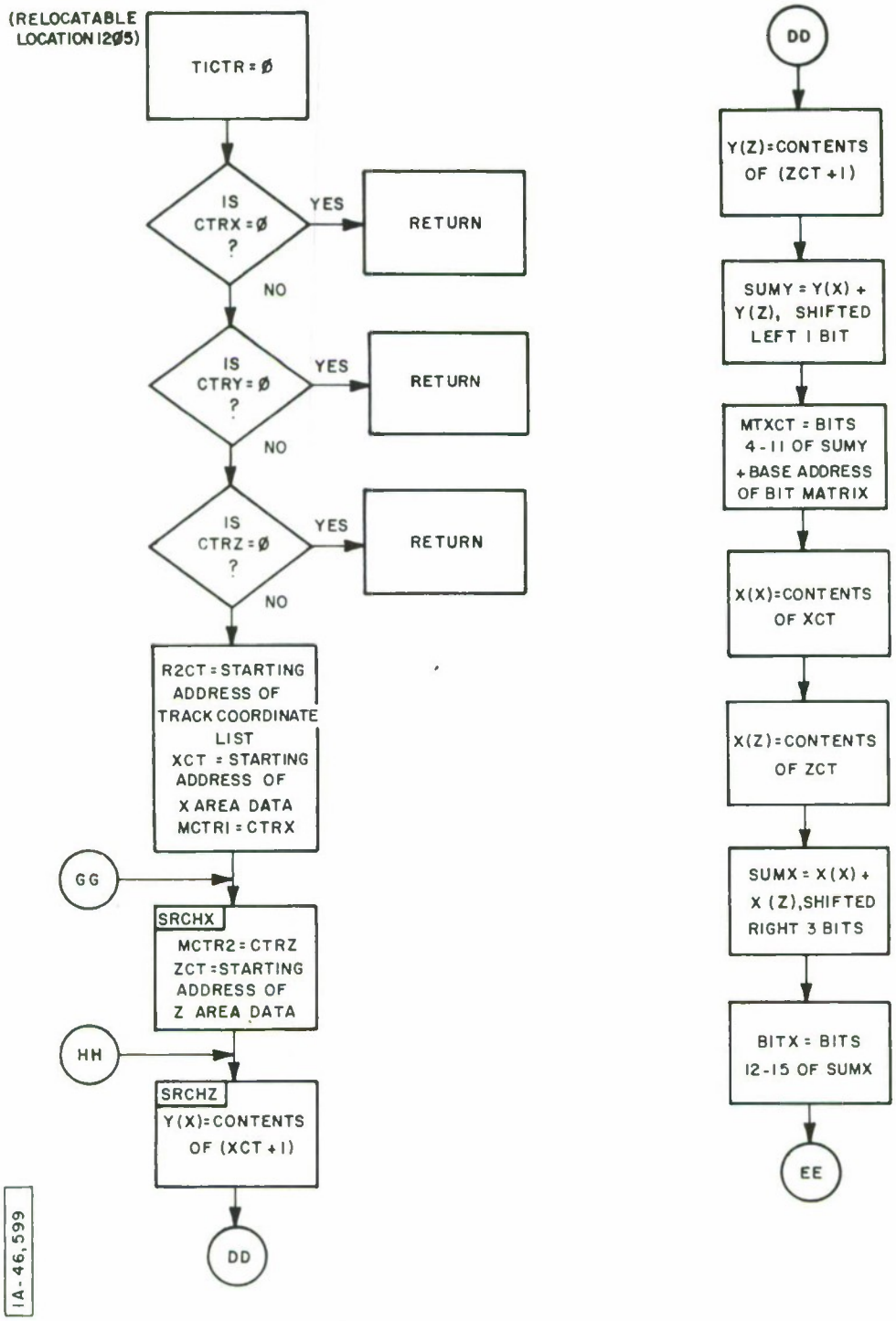


*

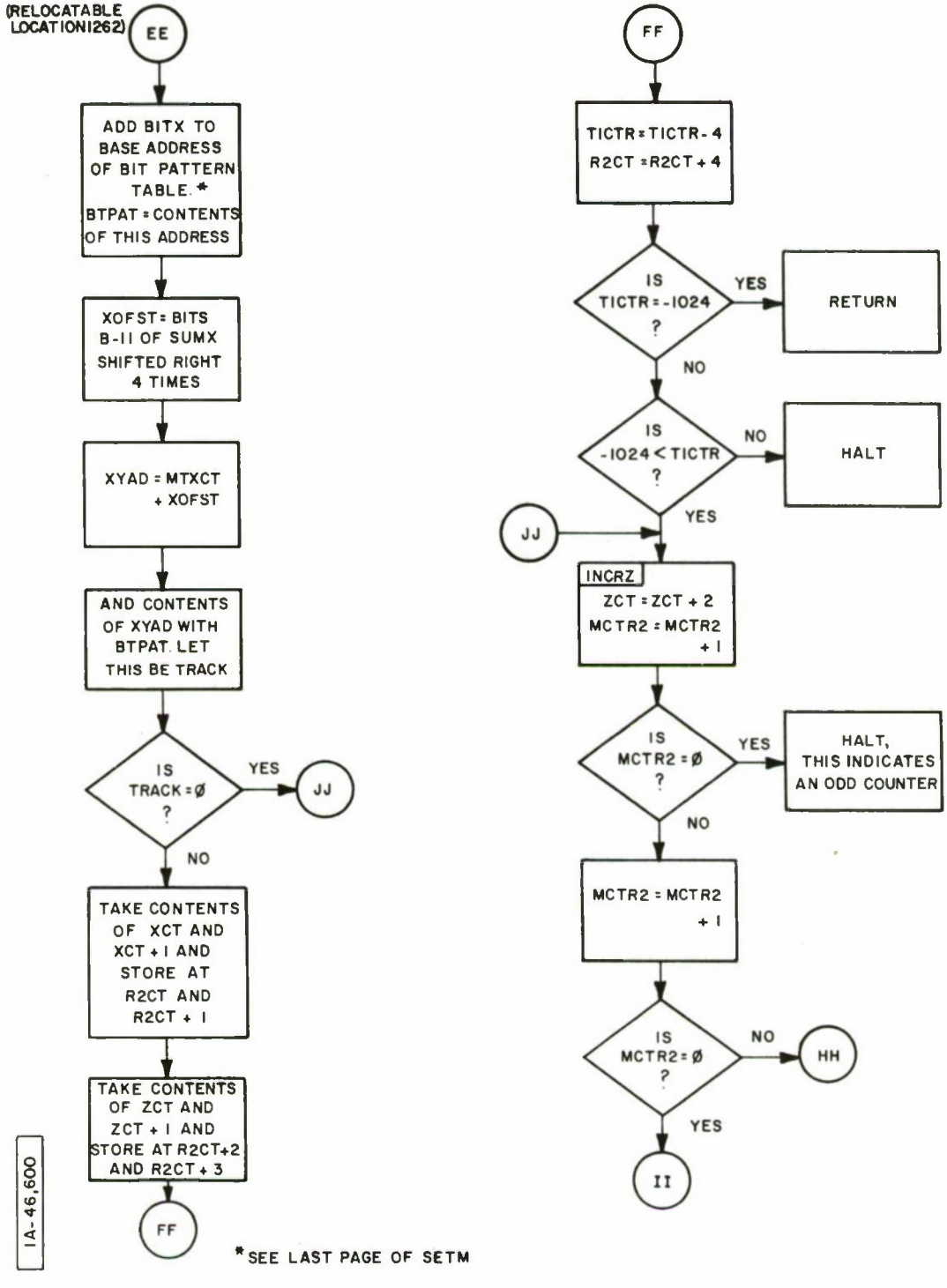
BIT PATTERN TABLE
1 Ø Ø Ø Ø Ø
Ø 4 Ø Ø Ø Ø
Ø 2 Ø Ø Ø Ø
Ø 1 Ø Ø Ø Ø
Ø Ø 4 Ø Ø Ø
Ø Ø 2 Ø Ø Ø
Ø Ø 1 Ø Ø Ø
Ø Ø Ø 4 Ø Ø
Ø Ø Ø 2 Ø Ø
Ø Ø Ø 1 Ø Ø
Ø Ø Ø Ø 4 Ø
Ø Ø Ø Ø 2 Ø
Ø Ø Ø Ø 1 Ø
Ø Ø Ø Ø Ø 4
Ø Ø Ø Ø Ø 2
Ø Ø Ø Ø Ø 1

IA-46,595

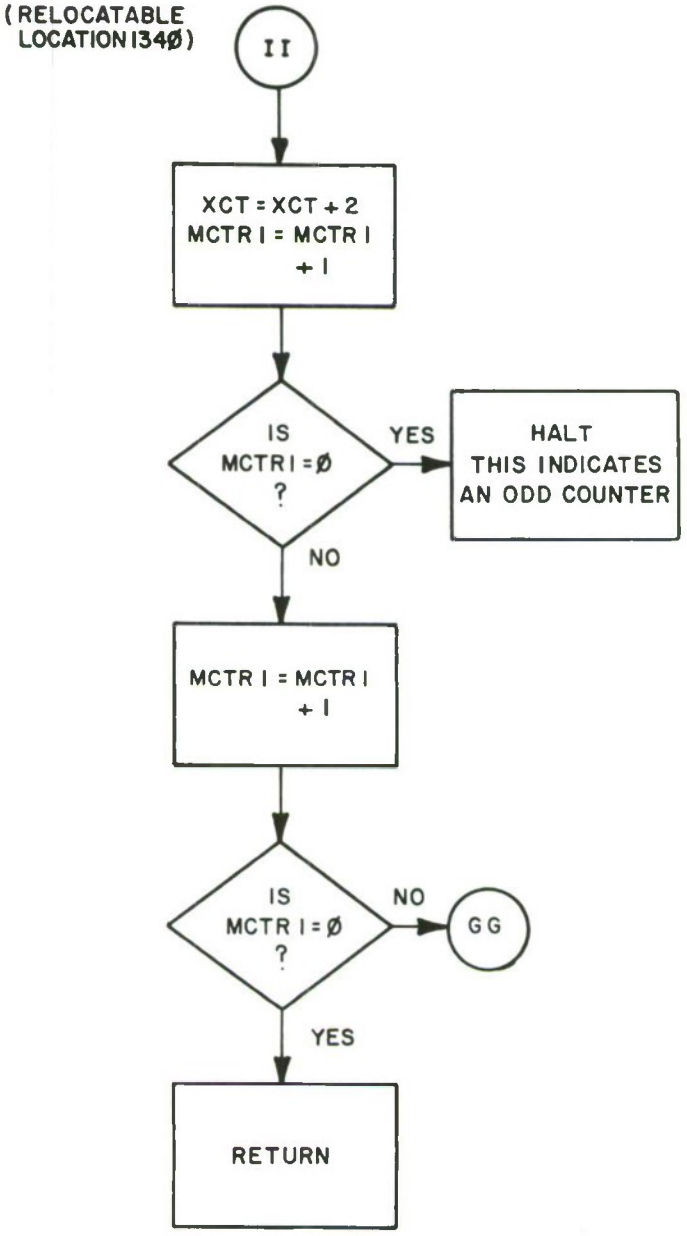
SETM - 2



CHCKM - SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 144)

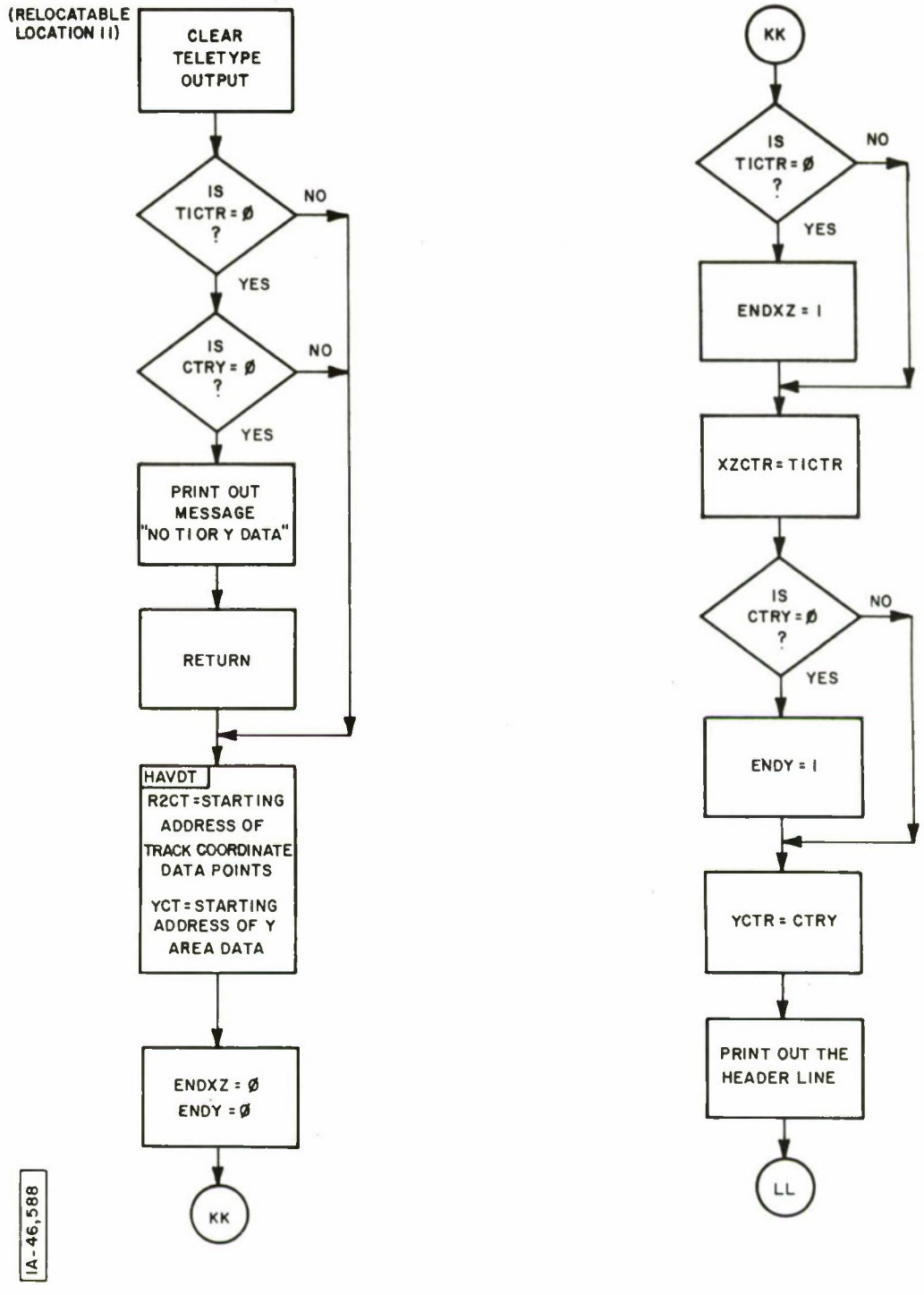

(RELOCATABLE LOCATION1262)
EE
ADD BITX TO BASE ADDRESS OF BIT PATTERN TABLE.*
BTPAT = CONTENTS OF THIS ADDRESS
XOFST = BITS B-11 OF SUMX SHIFTED RIGHT 4 TIMES
XYAD = MTXCT + XOFST
AND CONTENTS OF XYAD WITH BTPAT. LET THIS BE TRACK
IS TRACK = Ø ?
YES
JJ
NO
TAKE CONTENTS OF XCT AND XCT + 1 AND STORE AT R2CT AND R2CT + 1
TAKE CONTENTS OF ZCT AND ZCT + 1 AND STORE AT R2CT+2 AND R2CT + 3
FF
1A-46,600
*SEE LAST PAGE OF SETM
FF
TICTR = TICTR - 4
R2CT = R2CT + 4
IS TICTR = -1024 ?
YES
RETURN
NO
IS -1024 < TICTR ?
NO
HALT
YES
JJ
INCRZ
ZCT = ZCT + 2
MCTR2 = MCTR2 + 1
IS MCTR2 = Ø ?
YES
HALT, THIS INDICATES AN ODD COUNTER
NO
MCTR2 = MCTR2 + 1
IS MCTR2 = Ø ?
NO
HH
YES
II

CHCKM - 2

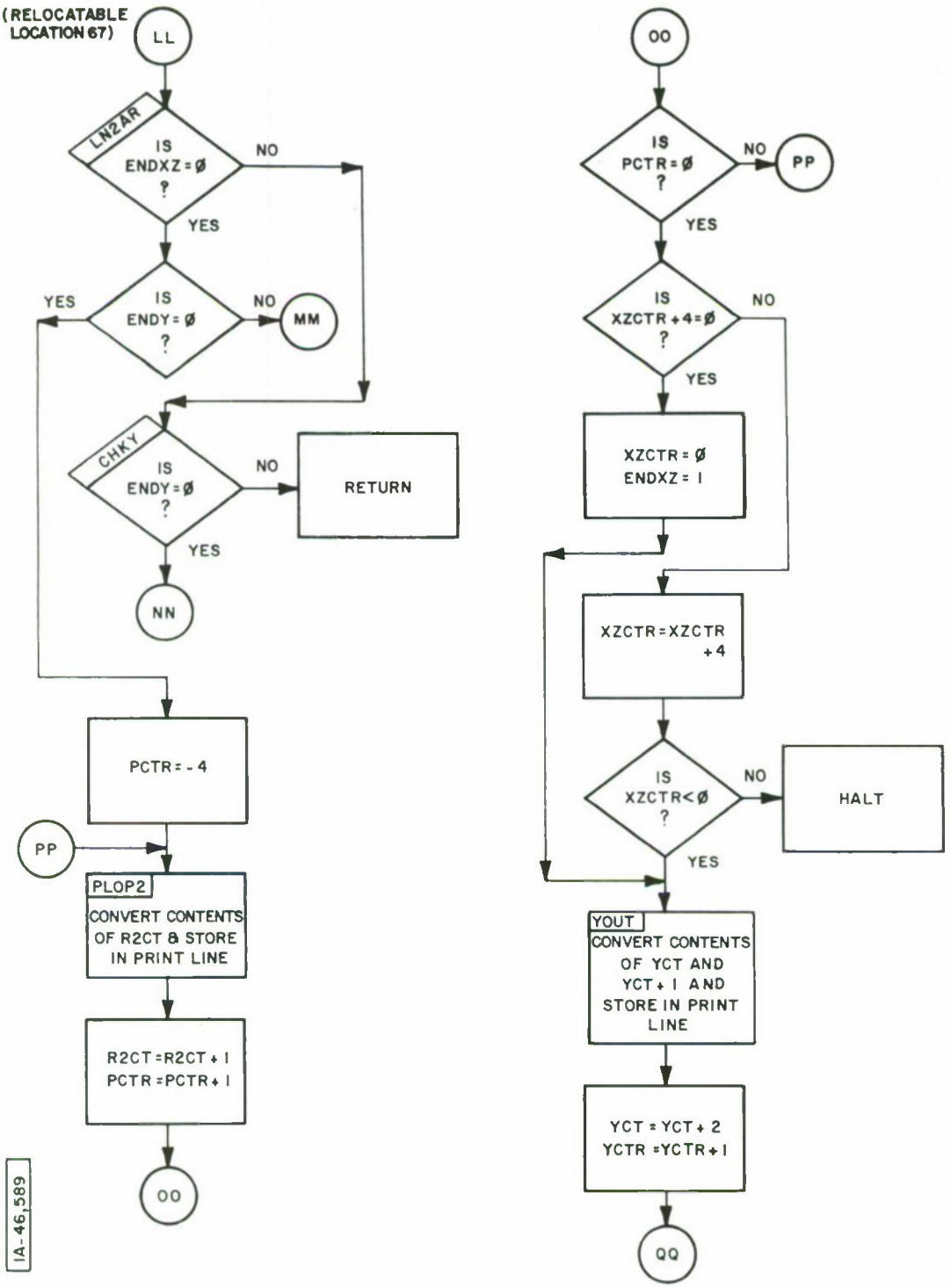


IA-46,601

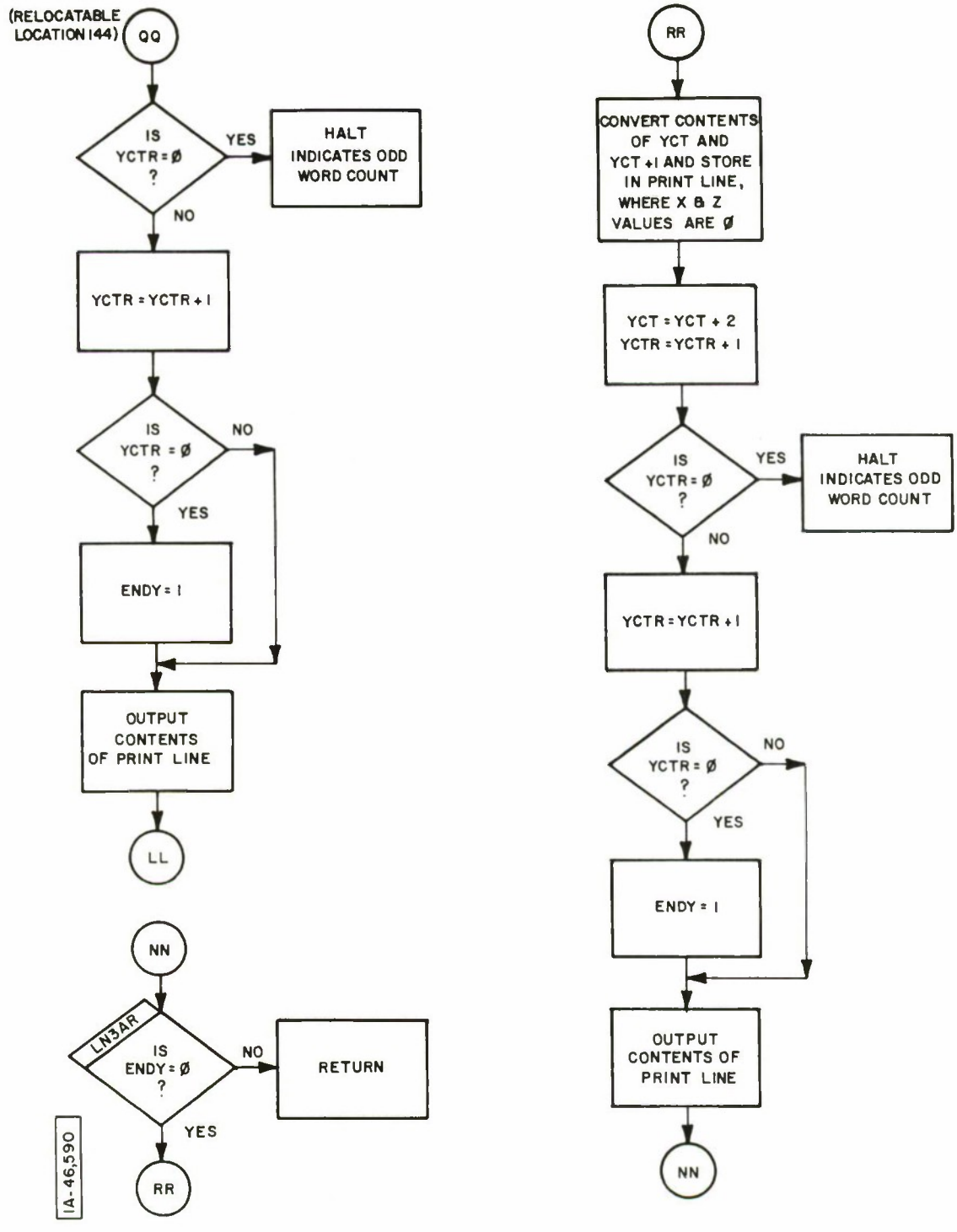
CHCKM - 3

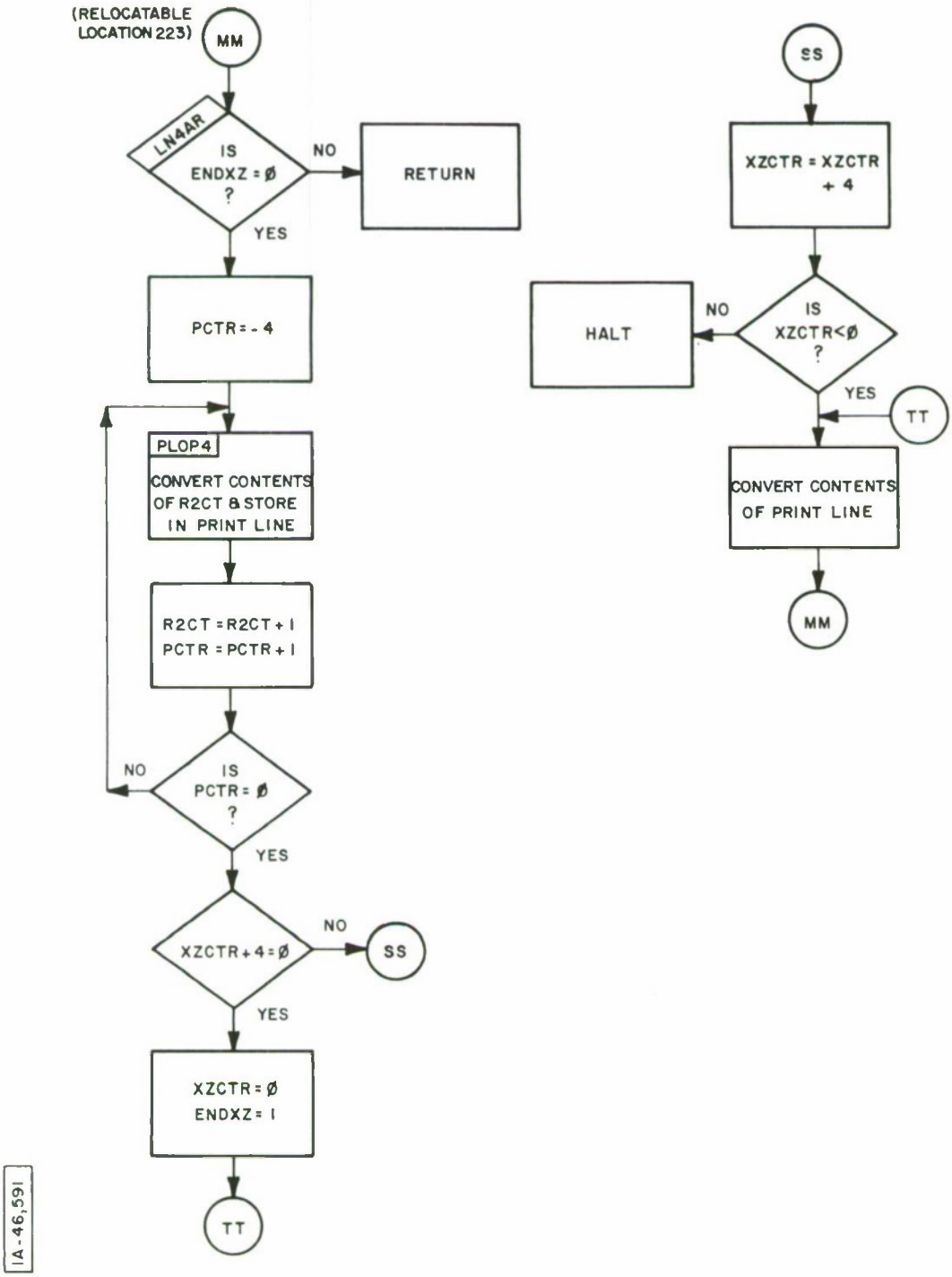


LSXYZ- SUBROUTINE OF TRKIN (LISTING STARTS ON PAGE 149)

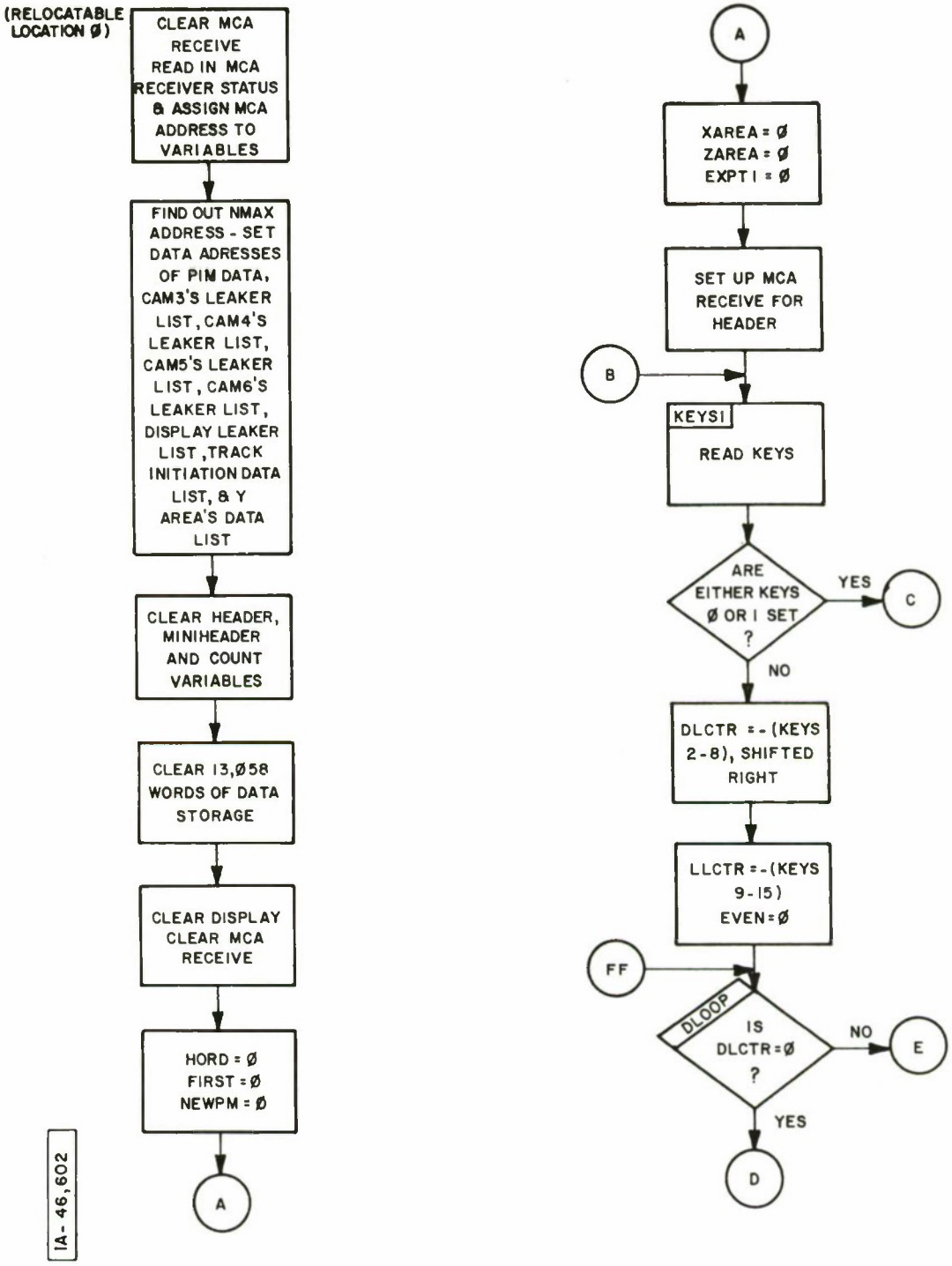

(RELOCATABLE LOCATION 67)
LL
LN2AR
IS ENDXZ = Ø ?
NO
YES
IS ENDY = Ø ?
YES
NO
MM
CHKY
IS ENDY = Ø ?
NO
RETURN
YES
NN
PCTR = - 4
PP
PLOP2
CONVERT CONTENTS OF R2CT & STORE IN PRINT LINE
R2CT = R2CT + 1
PCTR = PCTR + 1
OO
IA-46,589
OO
IS PCTR = Ø ?
NO
PP
YES
IS XZCTR + 4 = Ø ?
NO
YES
XZCTR = Ø
ENDXZ = 1
XZCTR = XZCTR + 4
IS XZCTR < Ø ?
NO
HALT
YES
YOUT
CONVERT CONTENTS OF YCT AND YCT + 1 AND STORE IN PRINT LINE
YCT = YCT + 2
YCTR = YCTR + 1
QQ

LSXYZ - 2

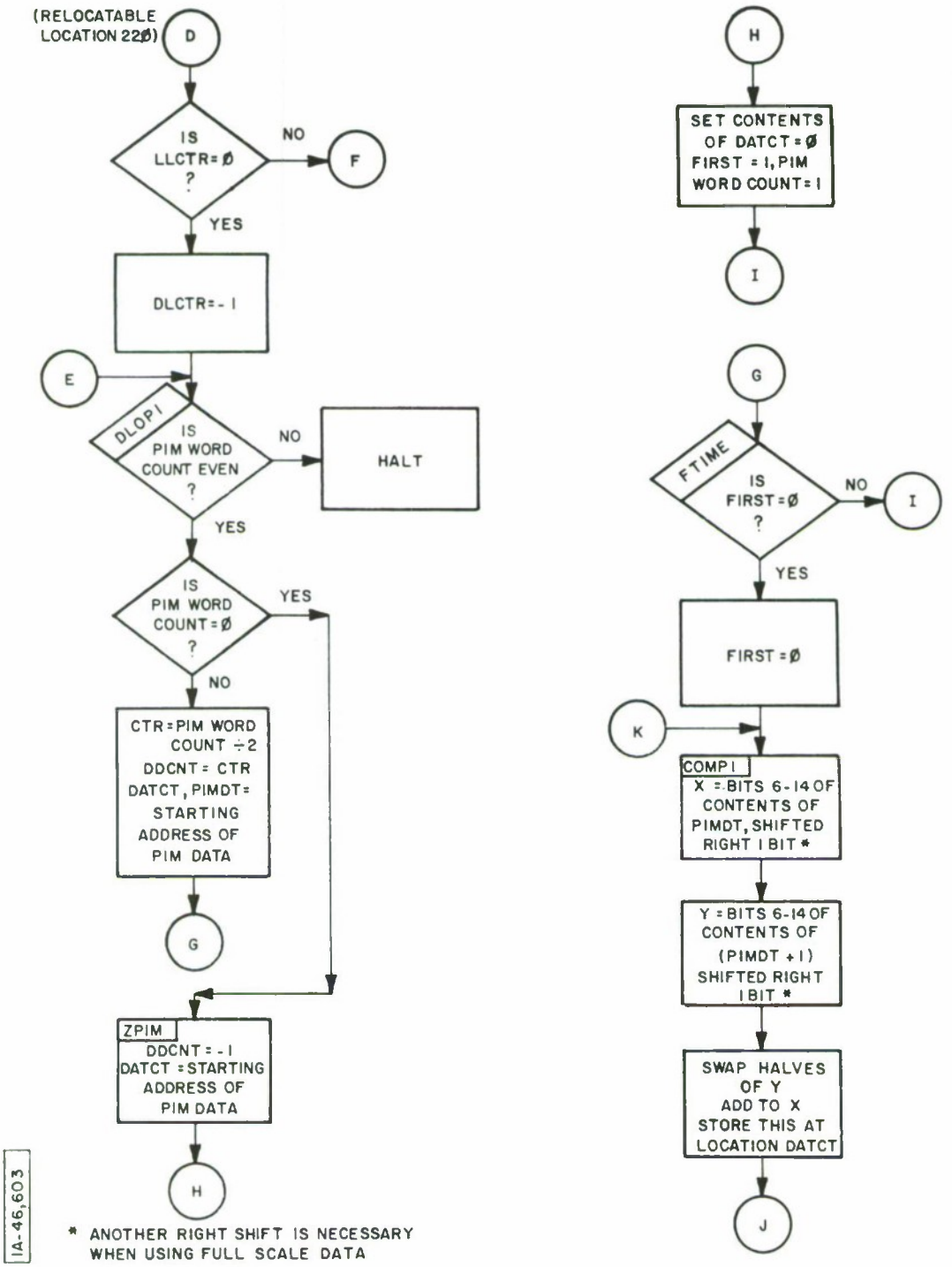

(RELOCATABLE LOCATION 144)
QQ
IS YCTR = Ø ?
YES
HALT INDICATES ODD WORD COUNT
NO
YCTR = YCTR + 1
IS YCTR = Ø ?
NO
YES
ENDY = 1
OUTPUT CONTENTS OF PRINT LINE
LL
NN
LN3AR
IS ENDY = Ø ?
NO
RETURN
YES
RR
IA-46,590
RR
CONVERT CONTENTS OF YCT AND YCT +1 AND STORE IN PRINT LINE, WHERE X & Z VALUES ARE Ø
YCT = YCT + 2
YCTR = YCTR + 1
IS YCTR = Ø ?
YES
HALT INDICATES ODD WORD COUNT
NO
YCTR = YCTR + 1
IS YCTR = Ø ?
NO
YES
ENDY = 1
OUTPUT CONTENTS OF PRINT LINE
NN


(RELOCATABLE LOCATION 223)
MM
LN4AR
IS ENDXZ = Ø ?
NO
RETURN
YES
PCTR = - 4
PLOP4
CONVERT CONTENTS OF R2CT & STORE IN PRINT LINE
R2CT = R2CT + 1
PCTR = PCTR + 1
NO
IS PCTR = Ø ?
YES
XZCTR + 4 = Ø
NO
SS
YES
XZCTR = Ø
ENDXZ = 1
TT
SS
XZCTR = XZCTR + 4
HALT
NO
IS XZCTR < Ø ?
YES
TT
CONVERT CONTENTS OF PRINT LINE
MM
IA-46,591

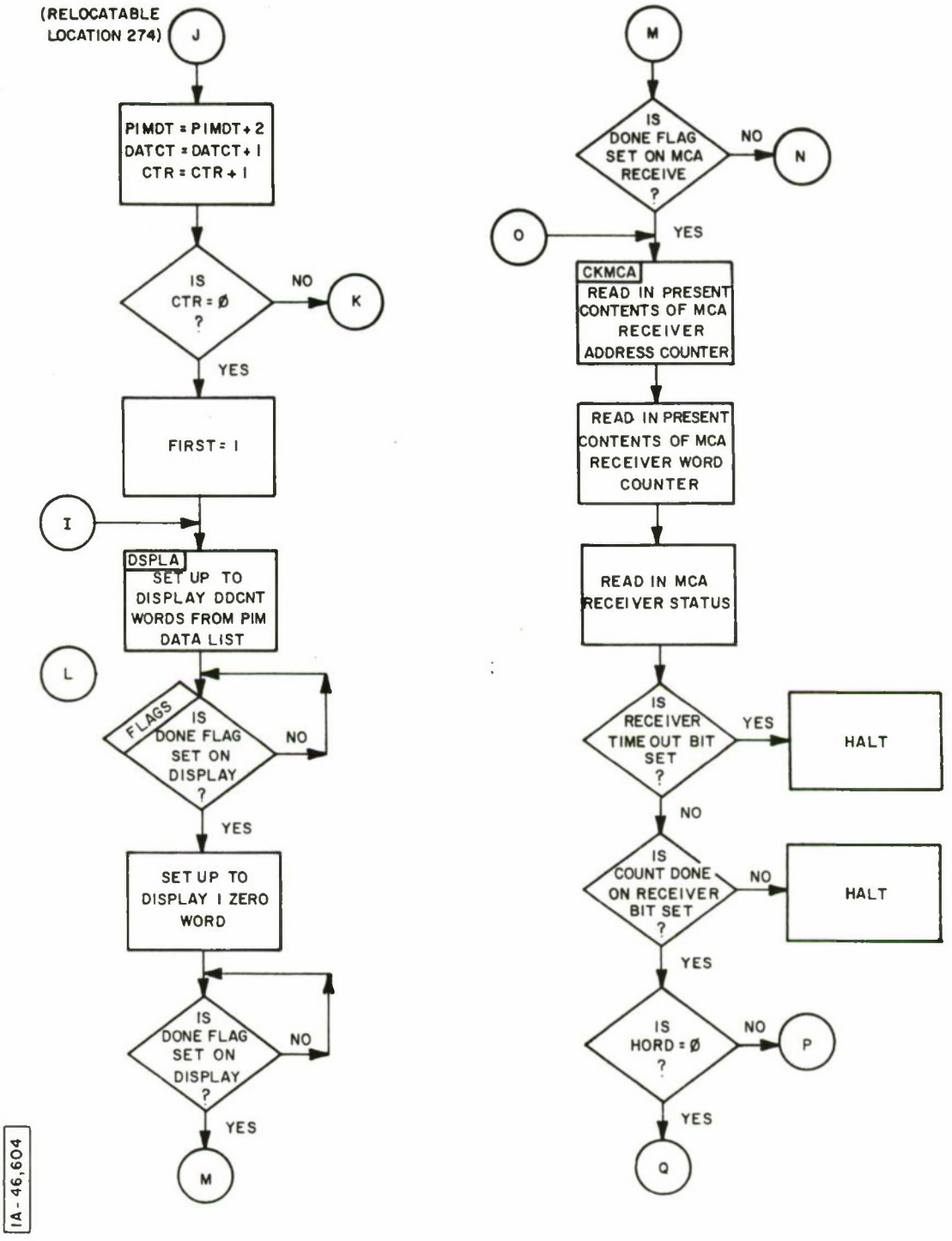
LSXYZ-4



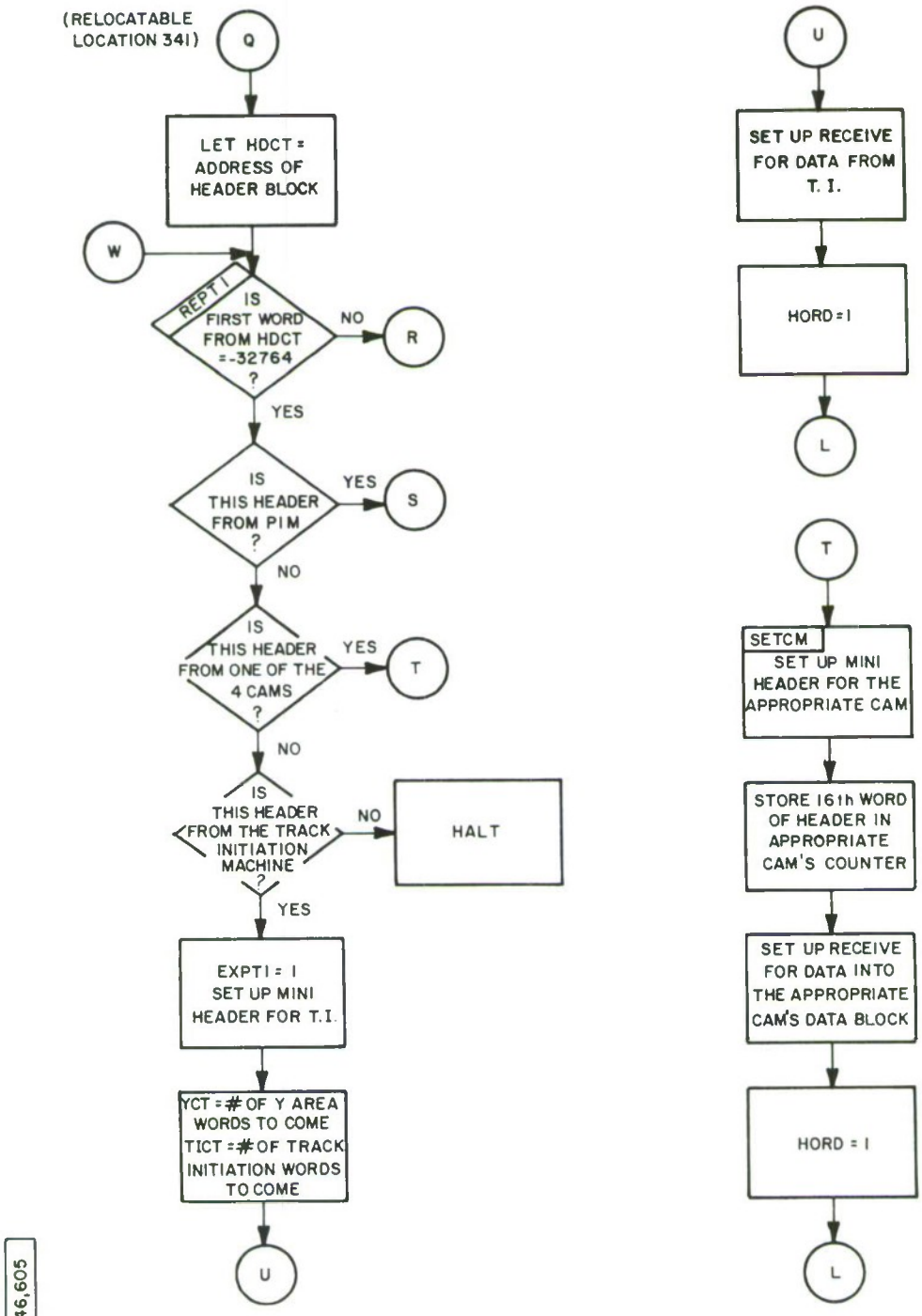
DISPLAY RECM - FOR DEMONSTRATION (LISTING STARTS ON PAGE 158)



DISPLAY RECM-2

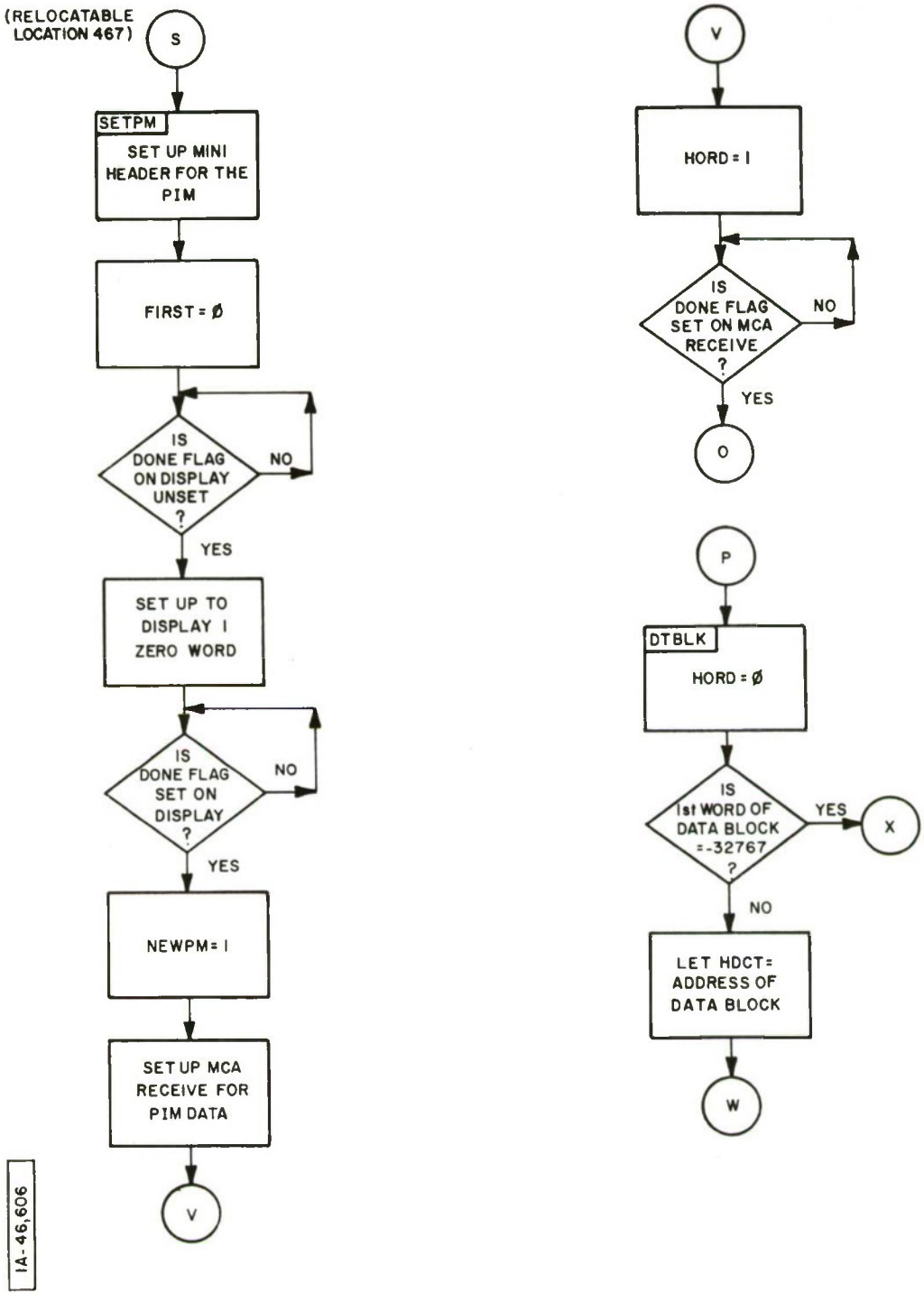


DISPLAY RECM - 3

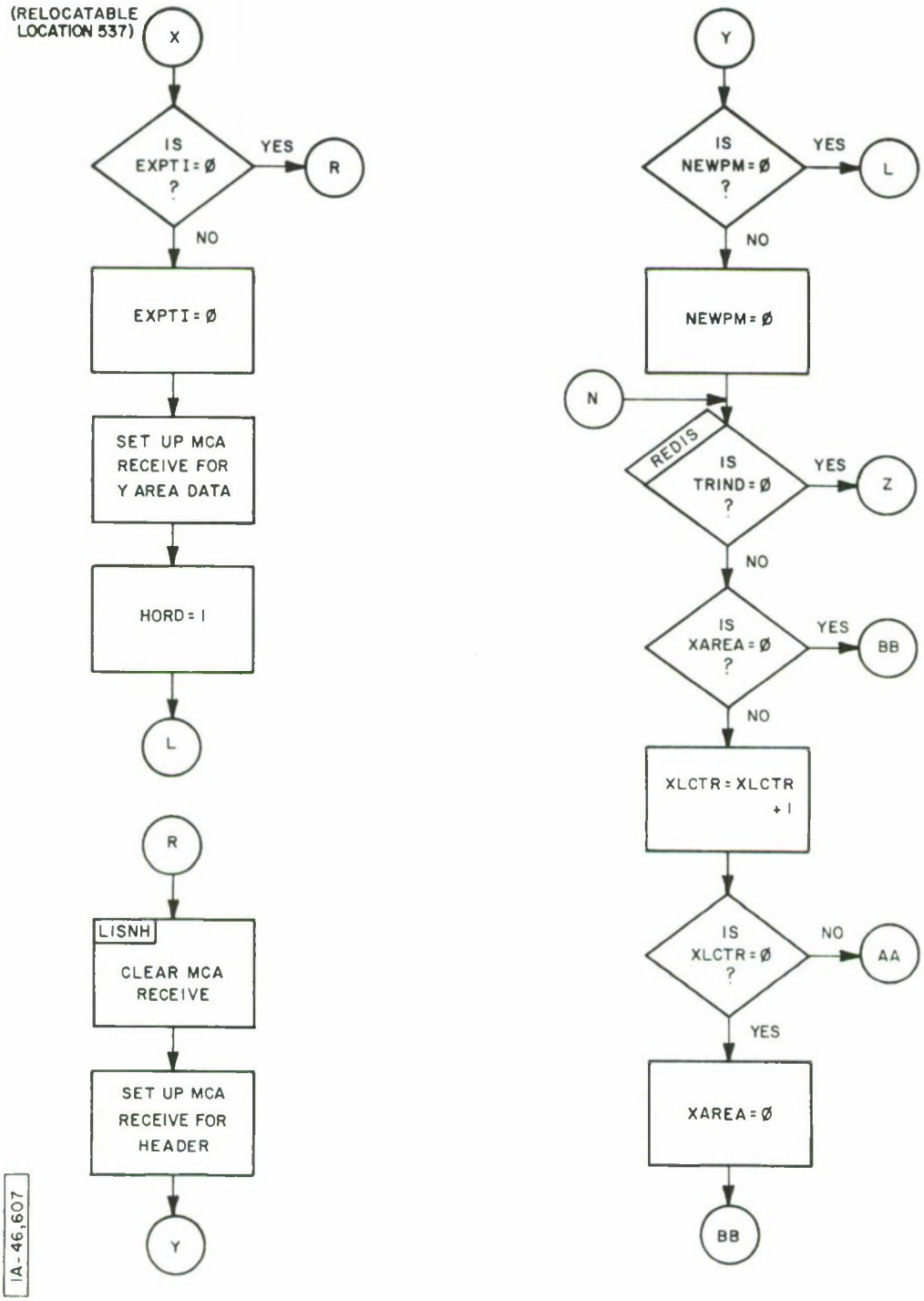


IA-46,605

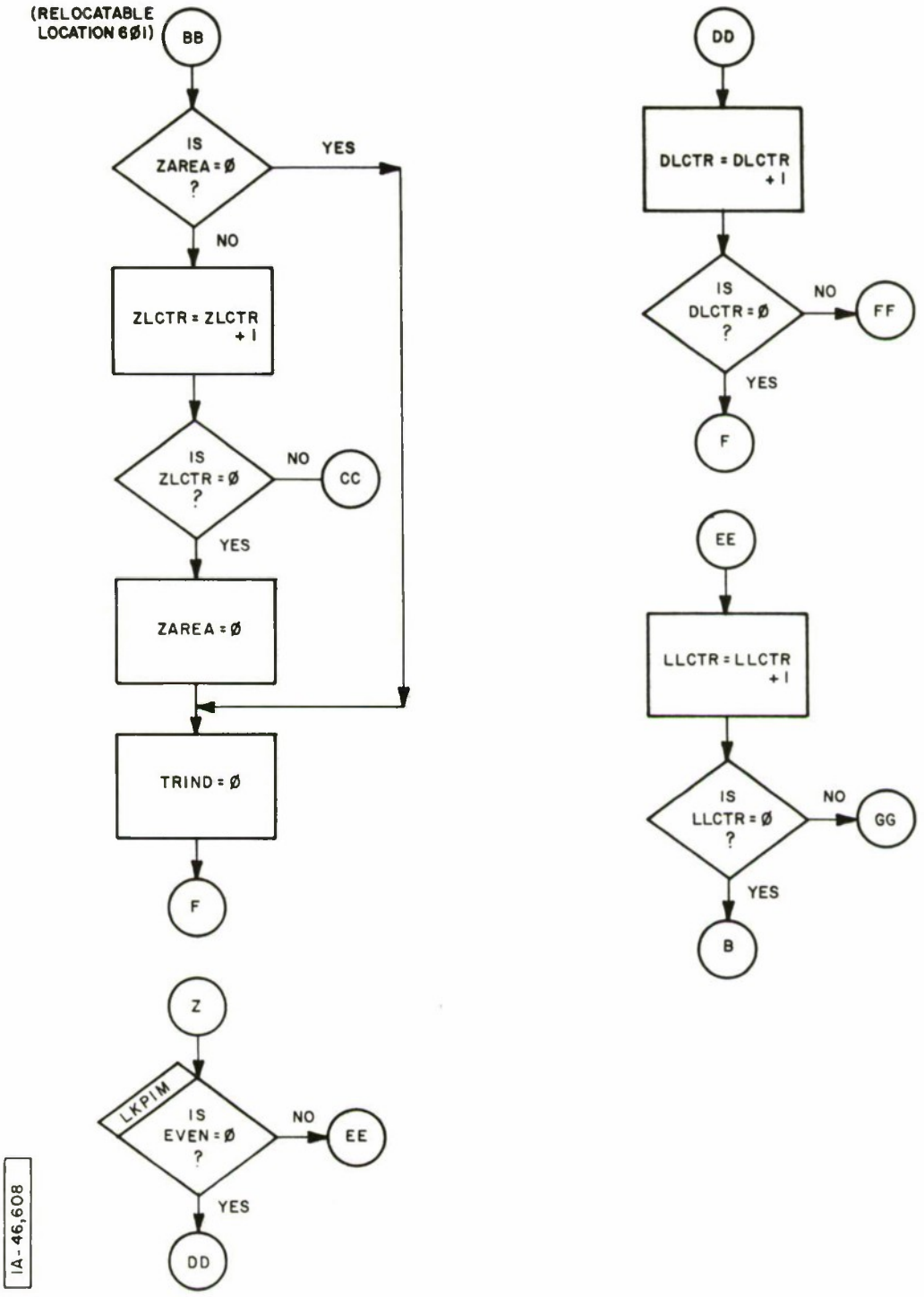
DISPLAY RECM - 4



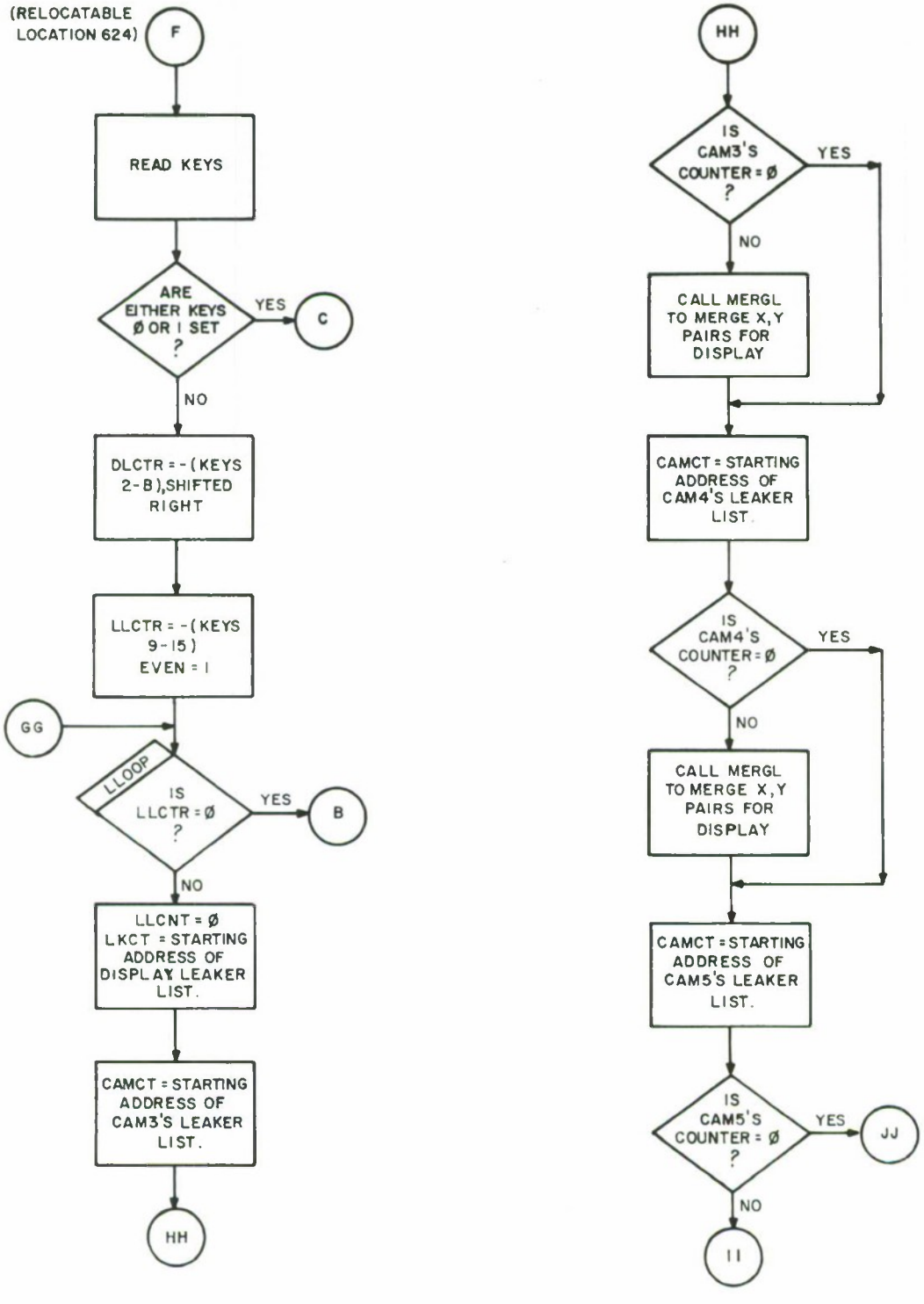
DISPLAY RECM-5



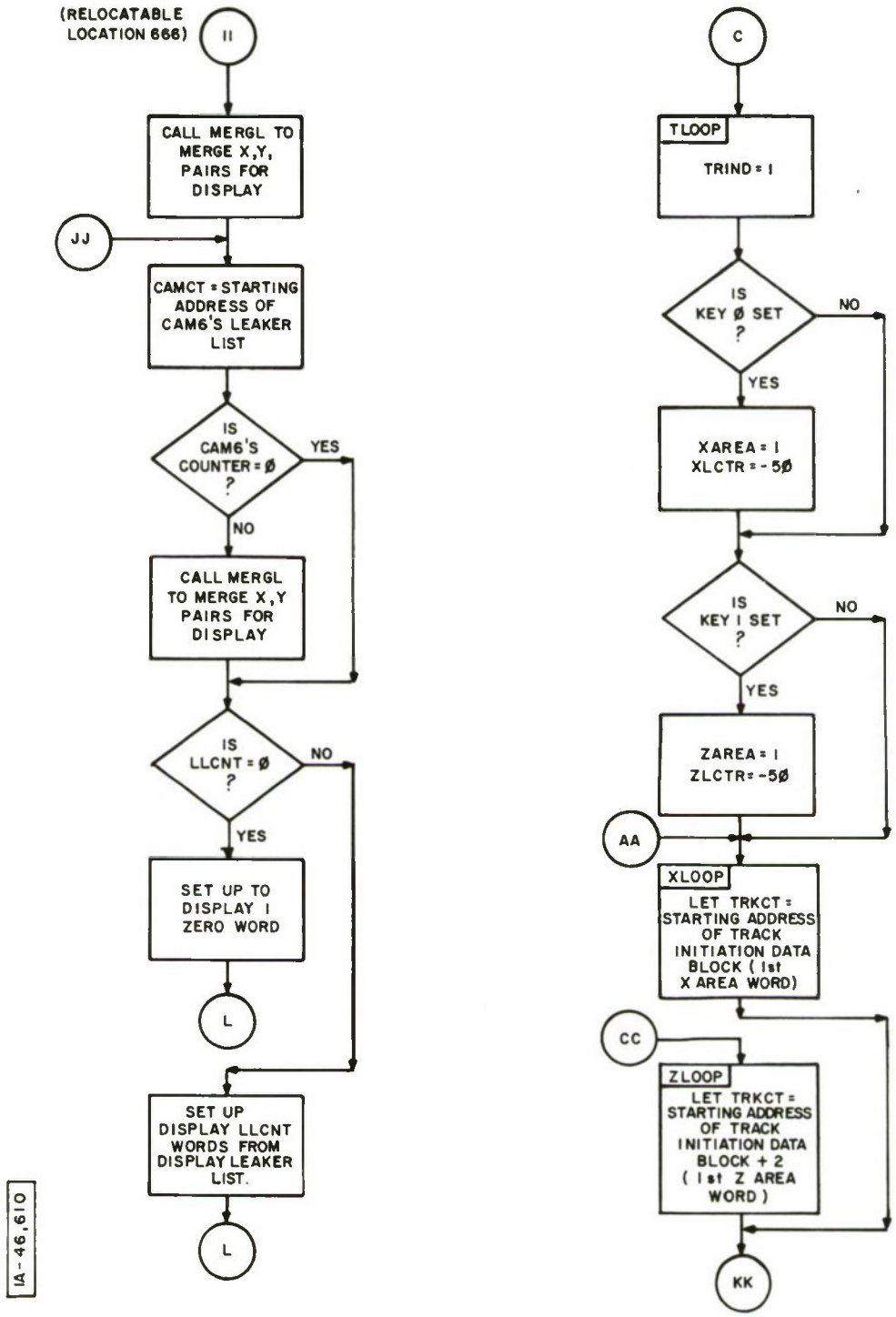
DISPLAY RECM - 6



DISPLAY RECM - 7

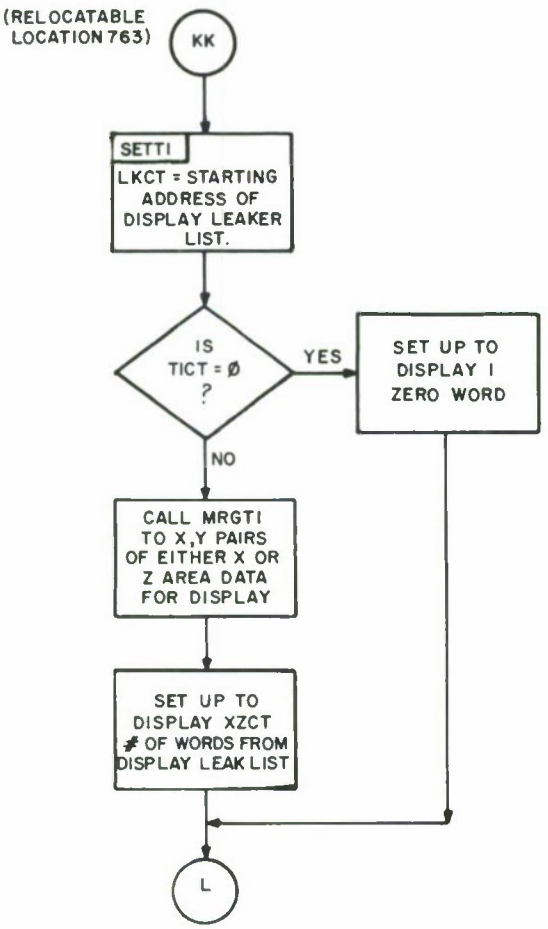


DISPLAY RECM-8



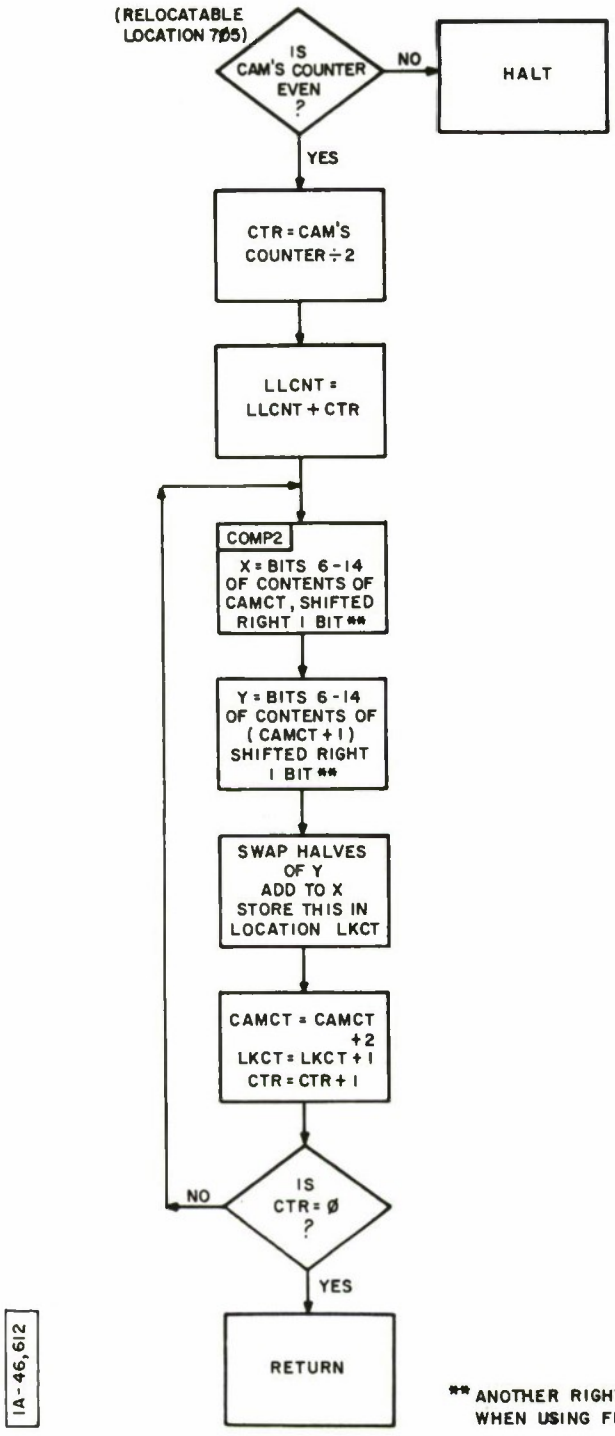
IA-46,610

DISPLAY RECM-9

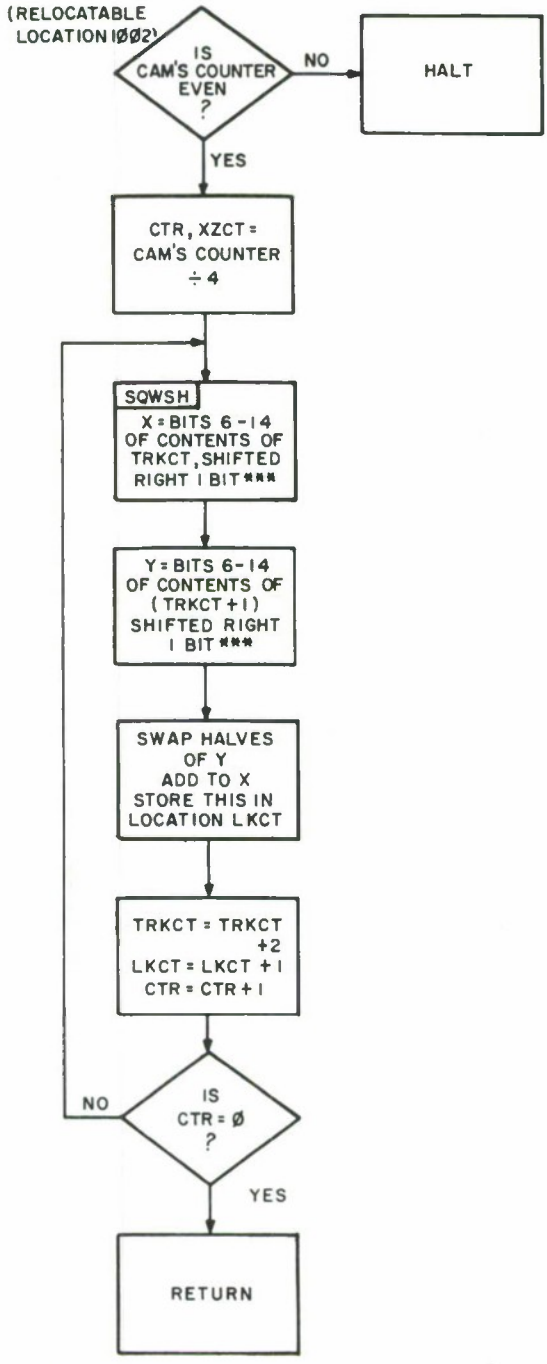


IA-46,611

DISPLAY RECM - 1Ø



MERGL - SUBROUTINE OF DISPLAY RECM (LISTING STARTS ON PAGE 168)



1A-46,613

*** ANOTHER RIGHT SHIFT IS NECESSARY WHEN USING FULL SCALE DATA

MRGT1 - SUBROUTINE OF DISPLAY RECM (LISTING STARTS ON PAGE 169)

APPENDIX III

CONTROL, DATA, AND PROGRAM BLOCK DESCRIPTIONS

A. Standard header block of 16 words

Word 1 = 32764., the header code

Word 2 = MCA address of sending computer, or -1 if this is reset header. (See description of future reset header).

Word 3 = Negative word count of data block or program which follows

Word 4 = 0 if data block follows

1 if program follows

Word 5 = 0 if data block follows, or the relative offset to entry point if program follows

Word 6 = Frame #

Word 7 = Scan # (currently the same as frame #)

Word 8 = Quadrant indicator

Word 9 = Disk Address Pointer for disk control block (currently unused)

Word 10-16 = unused

B. Reset header block of 16 words, to be used when polling logic is added to all programs. This logic will be added when the transmitter time out function is restored.

Word 1 = 32764., the header code

Word 2 = MCA address of the PIM, which is the only program to send reset headers

Word 3 = 0 if this is a polling reset header

1 if this is an assignment reset header

Note: If this is a polling header, the receiving machine will return a ready acknowledge block to the PIM indicating the MCA address and the function of this machine. If this is an assignment header, it will contain MCA addresses of the other machines in the system.

Word 4 =-1

Word 5 = DIM MCA address, if this is an assignment reset header

Word 6 = CAM1 MCA address, if this is an assignment reset header

Word 7 = CAM2 MCA address, if this is an assignment reset header

Word 8 = CAM3 MCA address, if this is an assignment reset header

Word 9 = CAM4 MCA address, if this is an assignment reset header

Word 10 = RECM1 MCA address, if this is an assignment reset header. Currently RECM1 = display RECM.

Word 11 = TRKIN MCA address, if this is an assignment reset header

Word 12 = RECM2 MCA address, if this is an assignment reset header

Currently RECM2 = display RECM

Words 13-16 - unused

C. Header block of 16 words, sent from CAMs to TRKIN and display RECM

Word 1 = 32764., the header code

Word 2 = MCA address of the sending CAM

Word 3 = Negative word count of leaker list

Word 4 = unused

Word 5 = Negative word count of PIM data block used in this cancellation

Word 6 = Frame # of PIM

Word 7 = Scan # of PIM (currently the same as frame #)

Word 8 = unused

Word 9 = Negative word count of PIM data block used in this cancellation

Word 10 = Frame # of DIM

Word 11 = Scan # of DIM (currently the same as frame #)

Word 12 = Disk Address Pointer of DIM (currently unused)

Words 13-16 = unused

D. Header block of 16 words, sent from TPKIN to display RECM

Word 1 = 32764., the header code

Word 2 = MCA address of sending computer

Word 3 = Negative word count of track data which follows

Word 4 = 0 because data block follows

Word 5 = Negative # of leakers from 1st CAM in relative frame 1

Word 6 = Negative # of leakers from 1st CAM in relative frame 3

Word 7 = Negative # of leakers from 2nd CAM in relative frame 1

Word 8 = Negative # of leakers from 2nd CAM in relative frame 3

Word 9 = Negative # of leakers from 3rd CAM in relative frame 1

Word 10 = Negative # of leakers from 3rd CAM in relative frame 3

Word 11 = Negative # of leakers from 4th CAM in relative frame 1

Word 12 = Negative # of leakers from 4th CAM in relative frame 3

Word 13 = MCA addresses, indicating order that CAMs' data blocks are in relative frame 1

Bits 0-3 = 4th CAM's MCA address

Bits 4-7 = 3rd CAM's MCA address

Bits 8-11 = 2nd CAM's MCA address

Bits 12-15 = 1st CAM's MCA address

Word 14 = Frame # of relative frame 1 data

Word 15 = Frame # of relative frame 3 data

Word 16 = Negative word count of # of leakers from 4 CAMs of relative frame 2 data. This data block will follow the track data block.

E. Mini header block of 4 words, used by display RECM for identification of current PIM data

Word 1 = Frame # from PIM header

Word 2 = Scan # from PIM header

Words 3-4 = unused

F. Mini header block of 4 words, used by display RECM for identification of current CAM data. The 4 CAM mini headers describe the 4 CAM's.

Word 1 = Frame # of PIM data used to generate the current leaker list

Word 2 = Scan # of PIM data

Word 3 = Frame # of DIM data used to generate the current leaker list

Word 4 = Scan # of DIM data

G. Mini header block of 4 words, used by display RECM for identification of current TRKIN data

Word 1 = Frame # of relative frame 1 data

Word 2 = Frame # of relative frame 3 data

Word 3 = Negative word count of relative frame 2 data which follows track data

H. Ready acknowledge block of 5 words

Word 1 = 32765., the ready code

Word 2 = MCA address of sending computer

Word 3 = Function type of sending computer. This code is used by the PIM to assign MCA addresses to the proper word in its reset assignment header.

1 = DIM function

2 = CAM function

3 = Display RECM function

4 = TRKIN function

Word 4 = 0 if header block just received was acceptable, 1 if header block just received was unacceptable

Word 5 = unused

I. Data or Program Blocks

Word 1 = 32767., if data block

32766., if program

Word 2 on = Data or program

BIBLIOGRAPHY

Data General Corporation, Extended Assembler User's Manual, 093-000040-01, Southboro, Mass. May, 1974.

Data General Corporation, How to Use the Nova Computers, DG NM-6, Southboro, Mass. October, 1972.

Data General Corporation, User's Manual RDOS Real Time Disk Operating System, 093-000075-05, Southboro, Mass., April, 1974.

The MITRE Corporation, The Multi-Minicomputer Processor, ESD-TR-75-351, Contract F19628-76-C0001, Bedford, Mass. September, 1975.